A Dynamic Risk Assessment Approach For Cooperative Medical Cyber-Physical Systems

Thesis approved by
the Department of Computer Science
Technische Universität Kaiserslautern
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)

to

M. Sc. Fábio Luiz Leite Júnior

Date of Defense:	11.06.2021
Dean of the Department:	Prof. Dr.-Ing. Jens Schmitt
Doctoral Committee:	
Head of the Doctoral Committee:	Prof. Dr. rer. nat. Klaus Schneider
Reviewer:	Prof. Dr.-Ing. Peter Liggesmeyer
Reviewer:	Prof. Dr. Ricardo João Cruz-Correia

D 386

Abstract

Medical cyber-physical systems (MCPS) emerged as an evolution of the relations between connected health systems, healthcare providers, and modern medical devices. Such systems combine independent medical devices at runtime in order to render new patient monitoring/control functionalities, such as physiological closed loops for controlling drug infusion or optimization of alarms. Despite the advances regarding alarm precision, healthcare providers still struggle with alarm flooding caused by the limited risk assessment models. Furthermore, these limitations also impose severe barriers on the adoption of automated supervision through autonomous actions, such as safety interlocks for avoiding overdosage. The literature has focused on the verification of safety parameters to assure the safety of treatment at runtime and thus optimize alarms and automated actions. Such solutions have relied on the definition of actuation ranges based on thresholds for a few monitored parameters. Given the very dynamic nature of the relevant context conditions (e.g., the patient's condition, treatment details, system configurations, etc.), fixed thresholds are a weak means for assessing the current risk. This thesis presents *an approach for enabling dynamic risk assessment for cooperative MCPS based on an adaptive Bayesian Networks (BN) model*. The main aim of the approach is to support continuous runtime risk assessment of the current situation based on relevant context and system information. The presented approach comprises (i) a dynamic risk analysis constituent, which corresponds to the elicitation of relevant risk parameters, risk metric building, and risk metric management; and (ii) a runtime risk classification constituent, which aims to analyze the current situation risk, establish risk classes, and identify and deploy mitigation measures. The proposed approach was evaluated and its feasibility proved by means of simulated experiments guided by an international team of medical experts with a focus on the requirements of efficacy, efficiency, and availability of patient treatment.

Acknowledgments

I would like to thank my family, especially my wife Isabella and my son Luis, for all the support, patience, comprehension, and care during the process. Moreover, I would also thank my mother Maria do Socorro and my brother Tiago for the support even though the physical distance between us, we always have strong bonds of mutual respect, collaboration, and love.

I would also like to thank Professor Liggesmeyer for the opportunity of developing my work in the Software Engineering Dependability (SEDA) Group of the University of Kaiserslautern. Further, I will always be thankful for the advices, support, management, patience, and promptitude to help me as a PhD candidate.

I am also grateful to my colleagues from the Fraunhofer IESE, especially Dr. Daniel Schneider, Dr. Rasmus Adler, and Dr. Pablo Antonino for their guidance, support, willingness, and incredible availability to help me with the development of my work. Likewise, I also want to thank all the colleagues from the Embedded Software Engineering and Embedded Systems Quality Assurance Department of the Fraunhofer IESE in the person of Jan Reich, Dr. Patrick Feth, and Christoph Dropmann. Special thanks go to Sonnhild Namingha who have provided me a special support during the doctorate studies in the Fraunhofer IESE as well as settlement in Germany, and linguistic issues.

I also want to thank my colleagues from the SEDA for the support and cooperation. Specially, Nikita Bhardwaj Haupt, Nishanth Laxman, Felix Möhrle, and Caroline Frey for the incredible availability and helpfulness.

Last but not least, I would like to thank the Brazillian government for the fellowship support and, specially, the Paraiba State University for providing me the professional opportunity of working abroad in one of the most important research centers of the world.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Problem Statement

Cooperative cyber-physical systems represent the cutting-edge evolution of embedded, adaptive, and open systems [24, 26, 164, 166, 231]. Such systems are capable of enabling diverse, and sometimes unpredictable, configurations through dynamic cooperation among different systems, from small devices to complex computer systems. Such systems have enabled innovative concepts from different fields such as Industrie 4.0 [127, 165], intelligent transportation systems and autonomous driving [41, 188, 242, 244], precision agriculture and smart farming [21, 47, 120], and the next generation of healthcare systems [46, 93, 216, 219].

Due to the dynamic nature of cooperative cyber-physical systems, they challenge the classic safety assurance approaches. Safety assurance methods and standards assume a complete understanding of systems and context to conduct the Hazard and Risk Assessment process [37, 95, 255]. To assure the safety of such dynamic systems, it is necessary to make worst-case assumptions (i.e., choose the worst possible scenarios of the cooperative CPS) for deriving the safety concept. Hence, this condition often results in less than optimal performance and thus compromises the overall operations of a system and its adoption [231, 233].

Runtime risk assessment methods have been considered as a potential solution provider for the stated problem. Such techniques are capable of updating the estimated risk of a deteriorating process according to the performance of the control system and its context factors. Although such techniques have advanced the state of the art for autonomy issues, they still struggle with dynamic composition. Moreover, the approaches for risk metrics derivation have been limited due to the insufficient completeness of their risk parameters and the fuzzy definition of situational spaces for establishing safety monitoring.

With that said, the core problem addressed in this thesis is: *The lack of dynamic risk assessment models that consider a wider set of relevant aspects of risk, such as system reconfigurations and context elements from the domain, have imposed severe limitations on safety assurance and performance for CPSoS.*

1.2 Solution Approach

To address the core problem targeted in this thesis, we designed a solution that shifts parts of the risk assessment process to runtime. This work provides an approach for guiding engineers through the derivation of dynamic risk assessment models, considering the impact of dynamic context and reconfiguration on risk metrics. Furthermore, to realize the proposed approach, we need to consider important issues, such as:

1. How to identify a relevant set of risk parameters in order to enhance the completeness of the risk metrics?

2. How to define a proper quantitative model for runtime risk assessment?

3. How to evaluate and implement the impact on the risk model caused by system adaptations?

4. How to effectively evaluate the risk in order to enhance safety and efficiency for the overall system behavior?

Therefore, this thesis introduces a model-based approach for the elicitation of risk parameters, a specification framework for deriving dynamic risk assessment so that models specified at design time can perform risk assessment based on the risk metrics and system configurations enabled at runtime. Finally, we provide guidelines for specifying runtime risk countermeasures to be enabled based on the current assessed risk of the situation.

1.3 Scientific Contribution

The overall requirement for runtime risk assessment is that a system must be able to systematically reason about its state (assess the current risk) so that it can estimate future states of the

monitored situation and the potential impact of potential system actions or modifications. Hence, in order to address the questions from section 1.2, we designed the following contributions:

Contrib. 1

A model-based approach for identifying and classifying relevant risk parameters. It defines models and methods that enable engineers to identify a relevant set of information that needs to be considered so that the risk metric can actually assess the current situational state. The runtime risk assessment process requires runtime updates of context and system variables so that the risk can be estimated. Such a model needs to aggregate multidisciplinary information from different domains of knowledge that no single role in the safety engineering process can be expected to have.

Contrib. 2

Furthermore, we needed to derive **guidelines for building the risk model based on all the relevant information organized by the provided model**. The risk model building process needs to consider particular aspects of both qualitative and the quantitative issues to derive efficient risk metrics. Moreover, a complete risk model needs to consider the impact of system adaptations on the risk model. In the provided approach, we therefore derived a method for defining risk metric management given the known abstract system adaptations.

Contrib. 3

An approach for enabling runtime risk classification and risk control specification. Since we provide an approach for enabling dynamic risk assessment for such CPSoS, it is also relevant to underpin it with reasoning about the acceptance levels and countermeasures for the current risk. As traditional approaches are focused on engineers evaluating the current risk level of such a system at design time, the mechanisms for risk control at design time can be different and often less effective than design time countermeasures. We therefore need to identify all the potential countermeasures available at runtime and then define a breakdown structure for situational risk analysis at runtime.

1.4 Validation of the Presented Approach

The contributions developed throughout this thesis have been validated in the field of connected healthcare systems using Medical Cyber-Physical Systems. In Chapter 7, we will describe how the approaches were validated in a simulated proof-of-concept project and an expanded experiment with different medical experts. In the simulated proof-of-concept experiment, we evaluated the feasibility of the risk metrics, the proposed risk monitoring architecture, and the risk management algorithm. Moreover, we expanded the evaluation process to a wider controlled experiment conducted with the support of experts from the medical domain, which defined four different evaluation scenarios. In the second expanded experiment, we validated the efficacy (with respect to the accuracy of the risk assessment) and efficiency (how fast the risk model can predict hazardous situations) of the approach, and how it can enhance treatment availability.

1.5 Thesis Structure

This thesis is structured into seven chapters. Following this introduction, we provide a literature overview and the state of the art in Chapter 2. The next constituent is the solution overview in Chapter 3, which presents a general understanding of the core contributions presented by our approach. Apart from that, a solution architecture is outlined that contains the building blocks required for the realization of the overall approach. Chapters 4 and 5 follow the structure outlined by the solution architecture and describe the Dynamic Risk Assessment approach. Afterwards, the evaluation of the presented approach is described in Chapter 6. Finally, Chapter 7 presents the conclusion, highlights of this thesis, and areas of future work.

2 Literature Review and Related Work

In this section, we present a literature review that provides the theoretical basis for understanding this thesis and evaluate state-of-the-art techniques in the field to highlight the relevance and the scientific contribution.

Several approaches have been proposed in the area of dynamic risk assessment, addressing a wide variety of from miscellaneous domains. The majority of the proposed techniques consider a specific set of risk parameters and are usually based on how this limited set of parameters affects the risk. However, due to the static nature of the applications and current standards, we have observed that in the literature, there is a lack of holistic approaches that consider not only some fundamental parameters usually related to the system or specific parts of the context.

In this sense, this chapter initially presents an overall survey on the current state of the practice regarding hazard analysis and risk management. Following that, we present the foundations of safety certification approaches and dynamic risk management in order to classify the related works and create a map of the state-of-the-art works in this field. Finally, we present a summary and our conclusions about the survey.

2.1 Medical cyber-physical systems: the next frontier of healthcare systems

The modern healthcare industry has been undergoing deep transformations following the advances of the new paradigm of cooperative CPS [64, 206, 229]. Particularly in the healthcare domain, new treatments have emerged from the combination of modern medicine, new classes of devices and IoT advances (expanded networks, biosensors, smart actuation devices), new data analysis technologies (such as Big Data, predictive and analytic methods), and novel smart control systems. Such advances have contributed to the evolution of care in hospitals wards, intensive care units, and personalized care (including

precision medicine, homecare treatment, and other approaches) [229].

In this context of modern smart control systems, cyber-physical systems (CPS) play a key role in the evolution of traditional control systems into a new generation of smart control systems for the medical domain [12, 46, 93, 118, 179, 212, 223]. CPS have been defined as the integration of computation with physical processes, where embedded computers control physical processes through continuous monitoring, communication networks, and actuators providing feedback loops [27, 128, 211, 232]. Due to their inherent dynamic and adaptive nature, CPS have challenged the safety-critical industry and certification bodies when it comes to assuring safety for such systems [232]. Currently, CPS are widely used in the fields of healthcare and medicine, transportation systems, agriculture, avionics, electric power, and others.

For the healthcare industry, medical cyber-physical systems (MCPS) are a fundamental driver for the new class of modern systems and treatments. According to I. Lee [129], MCPS are a combination of embedded software controlling the devices, networking capabilities, and complicated physical dynamics exhibited by patient bodies, which makes modern medical device systems a distinct class of CPS. This class of complex systems has enabled a plethora of new treatments and applications such as infusion control systems [173], telemedicine and home care monitoring [103], smart alarms and integrated control rooms in hospitals [116, 176, 221], clinical diagnosis and decision support systems [107], and so on. Although MCPS have contributed to the realization of several new classes of systems and applications, fundamental safety-related key issues still need to be addressed carefully by industry, academia, and certification bodies [118, 132, 223]. Therefore, this class of systems requires action in diverse areas. Among these, we focus on adaptiveness, autonomy, openness, cooperation, and safety.

The main aim of the interaction of MCPS with the physical world is to control and monitor physical processes through the sensors, actuators, and analytical modules of an MCPS. This expected behavior imposes adaptation requirements on MCPS to enable

them to respond to changes in the environment. As stated in [133], **self-adaptation** is the ability of systems to modify their behavior and/or structure in response to their perception of the environment and the system itself, and their goals. Hence, [122] categorized five aspects of adaptation, which we can use to analyze how a system behaves according to the changes in the environment that affect either the system or relevant elements of the context, namely:

- Reason – understanding the reason for the adaptation that leads the system to adapt. It can be trigered by changes in the technical resources (hardware or software defects; alternative network connection); changes in the environment (state of context variables); or changes regarding the users (composition of the user group).

- Adaptation control – regarding the planning and implementation of adaptation mechanisms. The main aim is to define how the adaptation protocols will be performed by the system entities. Goals, rules, and system policies are specified to define models to control the adaptation.

- Time – referrs to the right moment to adapt. It is possible to define a preditive adaptation approach that anticipates the changes that impact on the system and that must identify the need for adaptation before a decrease in performance occurs. The alternative are reactive approaches, which trigger adaptation after the system has perceived a change.

- Technique – reasoning about which extent of adaptation is needed to respond to changes. This may impact on the configuration parameter level, the architectural structure, and contextual elements.

- Level – identifying where the adaptation logic should be implemented. In this sense, we need to consider different implementation levels such as application, middleware or system software, communication network structure, technical resources, or context elements.

Kephart and Chess [113] defined a widely accepted reference model for organizing the realization of all the above elements. The MAPE-K model comprises four activities for managing adaptive behavior that are typical for a control loop:

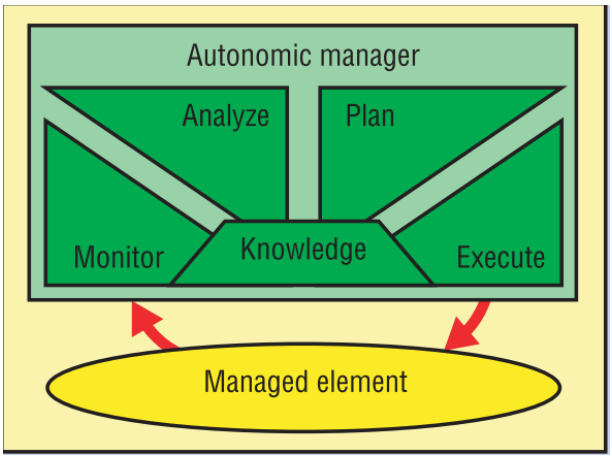


Figure 1: Structure of an autonomic element according to Kephart and Chess [113].

Figure 1 depicts different elements for the so-called autonomic manager. It is responsible for the **managed element**, which could be a referred resource that needs to be managed, as defined above. There are four activities: (1) *Monitor* refers to monitoring the system environment to gather relevant context information that might affect system properties and/or system requirements. (2) *Analyze* refers to analyzing the gathered information with respect to the necessity of self-adaptation. (3) *Plan* is about planning adequate actions to adapt the system in accordance with the results of the analyses. (4) *Execute* is related to the implementation of the adaptation plan. All these activities use a common knowledge base, which contains the adaptation information for the different activities and provides the basis for information management between the different activities.

Moreover, MCPS need to provide **autonomous** functions to realize either physical control functions or reconfiguration actions in order to adapt to environmental changes. This is strongly related to adaptiveness and autonomy. Autonomous systems refer to systems capable of operating in a real-world environment without any form of external control for extended periods of time

[100, 163, 209]. The latter functionality is due to the inherent complex design of MCPS, which can be composed of several different orchestrated systems. This brings a new class of complexity into these potentially safety-critical systems: hard to analyze can become not analyzable at all [6, 60, 132]. In this sense, such capability potentially brings even more challenges to safety assurance and traditional certification methods [232].

Complex MCPS are also required to operate in **open** environments in order to dynamically recombine themselves to provide new behavior. According to [101, 149, 171], systems-of-systems (SoS) represent the integration of a finite number of constituent systems that are independent and operable, and which are networked together for a period of time to achieve a certain higher goal. This definition perfectly matches the requirements of MCPS due to the inherent concept of independent monitoring (of multiparameter monitors, pulse oximeters, respiration monitors, and so on) and delivery devices gathering data from the patient and actuating on the patient in order to provide some kind of care (infusion and insulin pumps, pulmonary ventilator, pacemakers, etc.). Moreover, we might have different administrative ownership of the system elements. For example, homecare treatment [137, 148, 202] commonly uses sensors and actuators under the patient's home ownership (devices and network) combined with analyzing and decision-making systems from clinics and hospitals. Such systems therefore exhibit the natural characteristics of emergent behavior and dynamic cooperation to deliver the final care to the patient, providing higher-level services that cannot be provided by a single system alone.

2.1.1 Medical Cyber-Physical Architecture

There has been a remarkable evolution of the reference architectures over time, from the first works about MCPS requirements and reference architectures [12, 77, 179] to the modern advances in Health Connected Systems and the Internet of Medical Things. Although several solutions have been proposed, the overall set of requirements still needs to be fulfilled and further advancements are required. For instance, we can still observe clear demands in the areas of system and device interoperability, healthcare network, context awareness,

dependability, autonomy, certifiability, and so on. Therefore, an open gap for MCPS is an abstract reference architecture capable of adapting according to the dynamic demands of such systems.

In this sense, as a result of several [12, 126, 129] works from the literature, the ASTM published the standard "ICE standard F2761" - "Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) — Part 1: General requirements and conceptual model" [72] with the support of different institutions such as AAMI MDPNP and FDA collaborating in a working group. The ICE standard (F2761) has become a fundamental standard for new initiatives that benefit from the ICE architecture and clinical use cases (contained in F2761 Annex B).

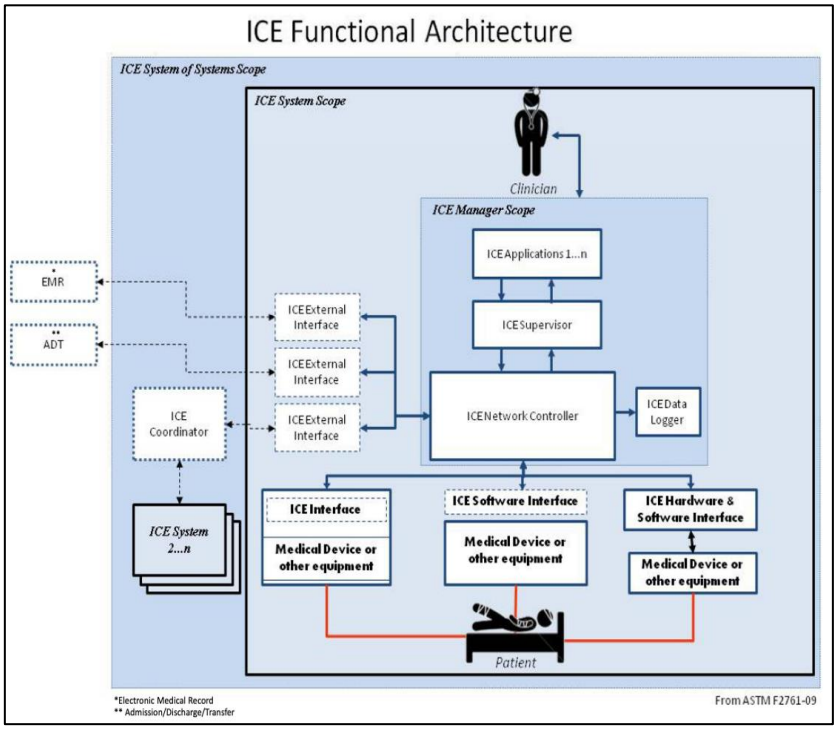


Figure 2: ICE Architecture [72]

In Figure 2, we present the proposed Integrated Clinical Environment – ICE standardization. First of all, we see the abstraction levels defined for the architecture: the ICE System of Systems Scope; the ICE System Scope; and ICE Manager Scope.

In the ICE Manager Scope, all the applications are defined that run in the context of the ICE Manager, such as smart alarms or physiological closed-loops. Moreover, in the ICE Manager Scope monitoring modules are defined that analyze either the system or network behavior to ensure that the functional capabilities, in accordance with the non-functional requirements in the device model of the ICE-compatible equipment, can be reliably delivered to the ICE supervisor.

In the ICE System Scope, all the elements relevant to the communication with the medical devices and related systems are defined. The following interfaces are shown in the figure: ICE Interface, ICE Software Interface, and ICE Hardware and Software Interface. They implement network protocols and communication applications that enable the communication with the ICE Network Controller. Moreover, the ICE external interfaces implement the communication with the external systems in the hospital.

In the System of Systems Scope, the modules that realize the communication with other ICE Managers or external systems are defined. Usually, ICE Managers enhance their quality of care when they access patient data from different sources such as electronic medical records from private and public systems, or medication history from drugstores. The ICE Coordinator can implement this interoperability and protocols that enable such communication.

The reference architecture defined in the ICE standard (F2761) specifies important modules, responsibilities, and roles for MCPS. It also defines requirements and design demands for such systems. Its relevance has been referenced by industry and by publications over time; for example, reference implementations [5, 33, 179] have been proposed and utilized by several commercial applications. However, considering MCPS as safety-critical systems, the standard still needs to define safety requirements related to the reference architecture as well as the relationship with risk assessment standards such as ISO 14971/ISO IEC 62304 / ISO IEC 60601.

Besides patient safety being a consolidated public concern, MCPS are part of a highly regulated industry. In this sense, there are restrictive demands for safety assurance argumentation and certification tasks for the manufacturers [16, 93, 223]. On the one hand, MCPS deliver fundamental functions to support and enable treatment for patients and increase the efficiency of healthcare providers. On the other hand, such functionalities are delivered by safety-critical functions that need to be analyzed, assessed, and evaluated according to the risk. We therefore need to identify all the demands from risk assessment techniques and standards in order to derive an approach that deals with all the challenges posed by MCPS for risk assessment.

2.2 Hazard Analysis and Risk Assessment for Safety-Critical Systems

Hazard and Risk Assessment (HaRA) establishes a set of practices and processes for safety-critical systems engineering quality assurance. As most safety-critical systems are found in highly regulated industries (for instance, avionics, rail transportation, energy, medical and health systems, nuclear and chemical power plants, automotive, and so on), the engineering process must comply with the requirements of the standards to enable manufacturers to launch their products on the market[54, 135, 170]. These industries are regulated by regulatory agencies responsible for reviewing and defining standards and regulations as well as by accreditation institutions that certify manufacturers, products, and processes so that they reach the market. Therefore, HaRA is a fundamental concern for the development of safety-critical systems.

The standard ISO IEC 61508 [252] defines a general understanding of Hazard and Risk Assessment (HaRA) as the process of identifying potential hazards and hazardous situations and estimating the risk of hazardous events determined through the analysis of all the reasonably foreseeable circumstances. In a practical way, ISO 31000 [181]establishes that risk assessment comprises a fundamental technical framework for the systemic analysis of the risk associated with a system’s overall behavior. The main task therefore is to structure the information and knowledge available at the detailed component/basic event level in order to assess the accident risk at the system level.

Particularly, for medical device systems [95], the risk management process comprises the whole lifecycle, including risk identification, analysis, evaluation, and control.

2.2.1 Risk Identification

The risk identification process aims to find those system usage conditions that might result in harm. Manufacturers need to establish the overall system behavior and context, and then identify hazards and hazardous situations where the system usage might lead to potential harm. Initially, manufacturers search accidents reports, literature, other standards, test results, and clinical evidence to identify potential harm that might affect users and caregivers [90, 95, 252].

For an effective risk identification, all the hazardous elements must be well defined for the stakeholders. According to IEC 61508, a hazard is defined as a "*potential source of harm*". This definition sounds rather vague and requires adding the concept of the "*state of a system and its environment*" [139] to enable a precise identification. This last definition then clarifies that the system behavior in its context might cause harm. The environment is defined by a foreseeable sequence of events and a resulting hazardous situation. Figure 3 summarizes all the terms. According to ISO 14971, a hazard is exposed during a sequence of events reaching a hazardous situation. This situation may then evolve into harm with a relative level of severity under certain probable conditions.

For example, consider this foreseeable sequence of usage for a defibrillator: (1) The battery of an implantable defibrillator reaches the end of its useful life; and (2) there is an inappropriately long interval between clinical follow-up visits. This can cause an omission failure due to the fact that the device does not deliver a defibrillation shock when arrhythmia occurs. Therefore, in such a situation, the patient might die due to this overall conjecture of system and context elements.

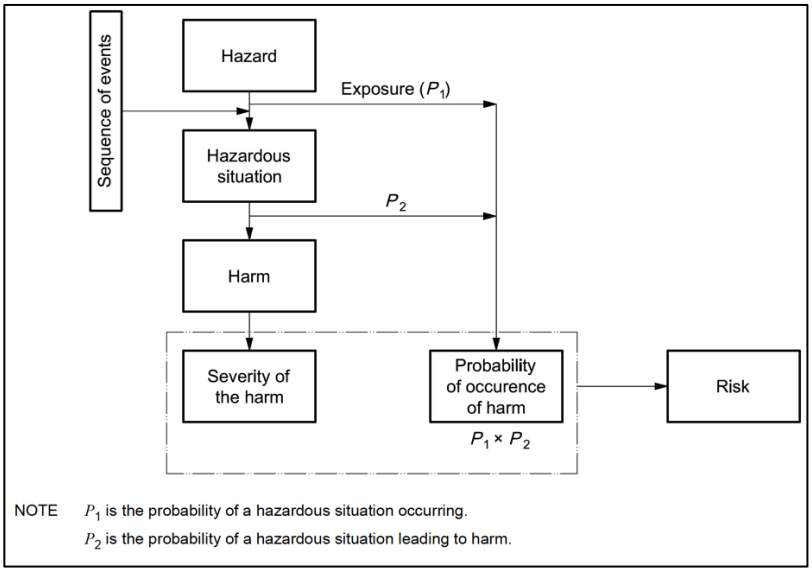


Figure 3 Risk definition according to ISO 14971

For medical devices, the severity of the harm might vary according to the treatment conditions. Severity is, in reality, a continuum; however, in practice, the use of a discrete number of severity levels simplifies the analysis. According to ISO 14971, the levels might range as in Table 1:

Common terms	Possible description
Catastrophic	Results in patient death
Critical	Results in permanent impairment or life-threatening injury
Serious	Results in injury or impairment requiring professional medical intervention
Minor	Results in temporary injury or impairment not requiring professional medical intervention
Negligible	Inconvenience or temporary discomfort

Table 1 Qualitative severity levels described in the ISO 14971

Traditional hazard analyses have been proposed to industry and literature for identifying hazards, hazard effects, and causal effects. The main aim of such techniques is to provide structured guidelines so that teams of engineers can systematically determine the system risk and thereby ascertain the significance

of hazards, making it possible to establish safety design measures for eliminating or mitigating the hazard. In this sense, the extent of the analyses must comprise the system, subsystem, facility, components, software, personnel, and their interrelationships.

According to [54, 191], there are more than 100 hazard analysis techniques that can be classified according to the time of their occurrence in the lifecycle, the applicable methods, and the extent of the analysis. Blueprints and standards [54, 95, 135, 170, 181, 253] call for a risk culture in industry, which is related to the fundamental role of risk management during all the development phases of the product and aims to avoid the classic mistake of considering risk management only in the latest development phases.

For the early phases, it is expected that if only few of the details of the medical device design are known, a preliminary hazard analysis (PHA)[184] will be carried out to identify the hazards, hazardous situations, and events that can cause harm. Using the standard PHA worksheet, hazards are identified inductively by asking "what if this component fails?" and hazards are also identified by asking how such an undesired event may happen.

Furthermore, in the later stages of the development, as the project is detailed with information about components and interactions, we can decide to use various techniques for analyzing components. For instance, Hazard and Operability Study (HAZOP) [8, 48, 57, 87] analysis defines guidewords for performing a brainstorming with engineers to identify causes and mainly consequences of system and component deviations. HAZOP is an inductive top-down approach, as the evaluation of system artifacts and usage scenarios starts from a high level of abstraction. Likewise, Failure Mode and Effects Analysis (FMEA) can be used to perform a systematic analysis of the components in order to look at a single-fault condition. This is done in a bottom-up manner, i.e., following the procedure to the next-higher functional system level.

Deductive techniques can also be applied to investigate the causes and consequences of the top events. Usually, they require more details from the system architecture because they provide more

refined information about the combination factors leading to an event, quantitative data, and potential consequences. Fault Tree Analysis (FTA) [89, 192] can be applied following a stepwise identification of undesirable system operations to successively lower system levels that will lead to the desired system level, which is usually either the component fault mode or the lowest level at which risk control measures can be applied. This will reveal the combinations most likely to lead to the postulated consequence. The results are represented pictorially in the form of a tree of fault modes. At each level in the tree, combinations of fault modes are described with logical operators (AND, OR, etc.). The fault modes identified in the tree can be events associated with hardware faults, human errors, or any other pertinent event that leads to the undesired event. They are not limited to single-fault conditions. Similarly, we can also use techniques such as Event Tree Analysis (ETA) [53] and Markov chain analysis [37, 68] to further refine aspects related to consequences and behavior over time.

Therefore, several techniques empower safety engineers to structure the safety-relevant information gathered from the experts regarding critical functions, deviations, hazardous events/situations, causal elements, and consequences. The next step is to collect all the relevant risk factors in order to derive risk metrics and then quantitatively assess the risk.

2.2.2 Risk Estimation

According to ISO 14971, risk analysis can be defined as the systematic use of available information to identify hazards and to estimate the risk. Therefore, based on the previous risk identification, the safety engineer must arrange and organize the gathered information to estimate the risk of the system performing a behavior considering the whole scenario description. Risk as a numerical quantity is thus useful for making decisions such as which risk prevention and mitigation measures to take, which measures to prioritize, and how to regulate risk acceptance.

To quantify the risk, we need to formally define the concept of risk. A well-known definition of risk is the combination between

the probability of occurrence of a harm and its severity (ISO 14971 and IEC 61508). Therefore, to derive the probability of occurrence of harm, we need to go through all the scenarios where hazardous situations might occur and look at their consequences to establish their severity. In [251], the author established the following formal definition:

$$Risk = \{\langle s_i, p_i(f_i, c_i)\rangle\}, i = 1, \dots, N + 1$$

Equation 1 Risk equation

where $s_i$represents the sequence of events of the i-th of N accident scenarios, f_i represents the frequency of occurrence of such a sequence of events, and c_i is the consequence that would result if that scenario were to occur. Therefore, the above definition leads to the identification of a complete set of scenarios. This set cannot be guaranteed, however, and thus is estimated by the joint probabilistic density function $p_i(f_i, c_i)$ describing the uncertainties in the frequency and consequences of the accident scenarios. The N+1 scenario is added to account for the set of scenarios unknown at the time of the analysis.

We can also observe that the formal definition of risk from ISO 14971 is completely in line with the definition in the Equation 1. In accordance with Figure 3, we can state the risk function as follows:

$$Risk = \{S_i \times (P1_i \times P2_i)\}, i = 1, \dots, N$$

Equation 2 Refined risk equation

where S_i represents the severity of the harm i, P1 the probability of the hazardous situation occurring, and P2 the probability of a hazardous situation causing harm i. The concept of scenario analysis is considered when we define P1 and P2. Therefore, this definition also expresses the dependency of the completeness of the scenario identification.

In this sense, further investigation of the literature revealed diverse risk metrics being utilized by different industries to calculate/estimate the risk. For example, the chemical and nuclear power plant industry [115, 248] often uses probabilistic risk

assessment techniques for specifying quantitative risk metrics in their risk monitors. Moreover, health systems have used probabilistic risk assessment and machine learning for descriptive and predictive risk assessment [106, 136, 220]. Hence, we established a characterization for classifying risk metrics based on aspects such as structure, technique, and adaptability, which will be presented next.

Risk Metric Structure

The risk metric structure defines how the parameters are arranged to process the risk assessment. It varies according to the application domain and the mathematical model used for risk estimation. For instance, in autonomous driving, there are several risk metrics based on the laws of physics that describe the relationship between the car and the other elements of the scenario, such as other cars, people, obstacles, etc. Hence, the risk metric structure for such cases is defined by the mathematical models that rule the laws of motion [15, 86]. Similarly, healthcare systems seek to model the processes of the human body and their relationship with interventions by the caregivers. These models are not totally precise due to the uncertainty inherent in the dynamicity of the scenario, which implies a lack of complete comprehension of the dynamicity of these processes and the consequent lack of precise models, especially for processes of the human body. Hence, we can establish the following classification for risk metric structures:

- Parameter threshold – A straightforward approach utilizes fixed thresholds defined by medical experts for defining risk classes; for example, blood saturation lower than 92%.

- Aggregated function – This structure relates different parameters for performing risk assessment. It may vary according to the technique (explained in the next section) and the risk model. The aggregation function can be expressed by a logic relation of parameters such as in [144, 154, 162] which are defined by invariant safety properties based on a previous hazard analysis specifying range values for these variables to derive the situation

status. Likewise, the risk metric can be expressed through a statistical model such as parameter-invariant (PAIN) monitoring [99, 241]. Moreover, probabilistic models can be applied to aggregate variables and identify the current system status, such as [105, 148, 174, 220, 228, 249]. All in all, risk metrics can be classified as predictive, descriptive, or evolutive.

Risk Metric Techniques

Risk metric techniques comprise sets of models and tools that can be utilized for modeling how the risk will behave over time. The risk model needs to accurately express how both the parameters and their relations will define the relation between the physical world and the system. We can thus classify these techniques as follows:

- Knowledge-driven – defines the risk model based on the application experts' knowledge. Normally, the selected model provides the structure and mechanisms for setting the risk model according to the experts' knowledge. Several models can be applied in such methods, e.g., Bayesian networks [82, 102], Dempster–Shafer theory [208], dynamic models (Markov chain models [68], or Petri networks [178]), and so on.

- Data-driven – defines automatic methods capable of building a risk model based on data. Recent developments in machine learning and analytics methods have supported the realization of such models. Such techniques have shown improved performance over competitive approaches when rich/dense training data is available [11, 79, 185, 221]. However, for several application domains, a consistent and reliable dataset is not available for the deployment of such risk models.

- Dual method – combination of the two methods to reinforce learning and experience. The learning approach suffers when historical data is not available or is not enough for deriving a reliable model. Such obstacles can be overcome with models that support expert knowledge

and can then be adjusted afterwards using evolutive techniques with the generated data [30, 39].

2.2.3 Risk Classification

According to ISO 14971, risk evaluation is a decision-making activity to classify whether the current functional risk fulfills the criteria previously defined and to assure that the residual risk is tolerable.

In the whole risk management process, the evaluation plays a fundamental role due to the definition of criteria, acceptability and tolerability issues. In this phase, the estimated risk for the safety functions must be evaluated and then, if the risk is not acceptable according to the criteria, risk control actions must be taken to reduce the risk to a tolerable or acceptable level.

Several techniques have been proposed to support the risk evaluation task. We will provide a brief description of ALARP, LOPA, and risk models based on graphs.

ALARP is a principle for risk evaluation that requires that any risk shall be reduced so far as is reasonably practicable [78, 91]. Assuming two extremes from tolerable and insignificant regions, if a risk falls between the two extremes and the ALARP principle has been applied, then the resulting risk is the tolerable risk for that specific application. The approach also defines risk classes based on the above definitions of acceptability and tolerability. In a qualitative way, the approach defines a relation between consequence and frequency to establish the following risk classes: Risk class I is an intolerable region; risk class II defines an undesirable but still tolerable region if risk reduction is impracticable or the costs are prohibitive; risk class III comprises a tolerable region where the cost for risk reduction would exceed the benefits; and risk class IV is in the broadly acceptable region. Therefore, the appropriate risk mitigation action must be taken according to the related risk class, and then safety requirements can be defined for the rest of the development process.

The Layers of Protection Analysis (LOPA) determines whether safety functions are required, as well as their respective safety

integrity level [32, 66, 91, 152, 224]. This risk analysis approach takes as input data developed in the hazard identification and utilizes the idea of systematic deployment of layers of protection for establishing barriers to avoid an accident. Hence, the safety requirements are defined with the proper SIL based on the established layers of protection.

The risk graph method enables a direct way for deriving the safety integrity level of a safety-related system from knowledge of the risk factors associated with the system. It is a semi-quantitative approach that establishes a relationship between the frequency and consequences of accidents through a graph. Moreover, further analysis might consider even more detailed information, such as frequency to exposure, possibility of avoidance, and probability of unwanted occurrence. This approach is adopted to simplify matters when a number of parameters are introduced which together describe the nature of the hazardous situation when safety-related systems fail or are not available. [91].

2.2.4 Risk control

Risk control aims to define a set of actions (or measures) to reduce the current assessed risk to the acceptable target defined during the evaluation phase, as we can see in the standards ISO 14971, IEC/ISO 61509, ISO 31000 [95, 181, 255].

The IEC 61508 standard places a strong focus on the specification of safety requirements, the respective derivation of the safety integrity level, and safety requirements allocation in order to achieve the required functional safety. To specify the overall safety requirements, it is necessary to analyze the hazardous events and thus specify a safety function for each of them. Further on, the target safety integrity requirement shall be specified for each safety function to meet the tolerable risk. The overall safety integrity requirements shall be specified in terms of either the risk reduction required to achieve the tolerable risk, or the tolerable hazardous event rate so as to meet the tolerable risk. Finally, the target failure measures and the associated safety integrity level must be allocated for each safety function to be carried out by a safety-related system. The allocation process is iterative and if it is found that the tolerable risk cannot be achieved, then the

specifications for the ECU control system, the designated safety-related systems, and the other risk reduction measures shall be modified, and the allocation repeated.

The main target of these measures is to mitigate the risk by reducing the probability of the occurrence of an accident or decreasing its severity. Hence, we can define the first classification for safety measures (or, using a more comprehensive definition, safety barriers), which are preventive actions against unwanted events and protective measures against unwanted outcomes [81, 91, 95, 214, 222]. Note that, whereas prevention tries to maintain the functioning of the system and to keep it going, protection does not need to do that. Indeed, protection may require the system to be shut down when a critical event occurs, as in the case of nuclear power plants, or that the normal functioning is reduced until the situation has returned to normal. This is acceptable because the goal in such cases is not the continued functioning but the safety of the overall system and its environment.

For example, ISO 14971 clearly states that the manufacturer shall use one or more of the following risk control options in the order of priority listed: a) inherent safety by design; b) protective measures in the medical device itself or in the manufacturing process; or c) information for safety. The medical device standards specify that preventive measures are the priority; however, protective measures are also strongly considered in several medical device projects.

Considering the "Bow-tie" model [152, 191], safety barriers (measures) are defined as physical or non-physical means intended to prevent, control, or mitigate undesired events or accidents [81, 114, 214]. Table 2 provides an overview of the application of risk barriers according to their application in the whole system lifecycle and highlights the type of safety barrier. Initially, prevention focuses on avoiding the causes that lead to a hazardous situation by strengthening the system development through fault tolerance techniques, quality assurance approaches, specification and implementation of safety requirements related to the product, and so on. Most of the prevention techniques are related to activities performed at development time, i.e., when the

system design can be modified and is under the manufacturer's control. However, such kinds of measures based on the system design can easily become outdated (sometimes even invalid or useless) because of the challenging dynamicity of MCPS.

The aim of mitigation barriers is to protect the targets from harm once an unwanted event has already happened. Usually, they are defined as the last contention barriers for mitigating the consequences of an event, and they must work independently and concurrently to prevent and control measures. Such measures can be implemented physically, such as shielding walls against radiation discharged by an X-ray machine, or defined by standards and human operators, such as clinical procedures for recovery of a patient in the event of overdosage. Since these kinds of barriers are therefore defined by a wider set of safety analyses comprising the whole organization, and since they are rarely controlled by the MCPS, we focus our definition of risk control measures on control safety barriers.

The function of control safety barriers is to prevent a transition from lack of control to loss of control [81, 191, 214]. This definition perfectly matches the concept of ALARP for dynamic and autonomous tasks performed by MCPS. Autonomy and dynamicity challenge safety assurance at design time due to the unfeasibility of analyzing and assuring all potential system and context variations at design time. However, control safety barriers are specified by the current risk, as we need to identify the risk classes in order to be able to allocate each measure accordingly. The main examples are safety interlocks, autonomous functions, and alarms.

Function	Definition	Example	Allocation

Prevention	aims at suppressing all potential causes of an event by changing the design of the equipment or the type of product used.	Safety requirements as defined by IEC/ISO 61508	Development time
		Pre-defined adaptation models	Development time/ Runtime
Control	aims at limiting the deviation from a normal situation to an unacceptable one.	Safety Interlocks (SiS)	Development time/ Runtime
		Autonomous actions	Runtime
		Alarms	Runtime
		Dynamic adaptation	Runtime
Mitigation	aims to protect the environment from the consequences of an unwanted event that has occurred.	Operator actions	Beyond system boundaries
		Plant alarms	Beyond system boundaries
		Walls	Beyond system boundaries

Table 2 Allocation of safety barrier functions according to the product's lifecycle

In Table 2, the allocation of the safety barriers is analyzed according to the development lifecycle. Most prevention measures or inherent safety by design aims to avoid unwanted events through safety requirements specification and allocation. However, the adaptability, autonomy, and dynamicity of MCPS challenges this definition of safety measures due to the changes in the system function and consequently in the hazardous situations. Therefore, control safety barriers can, in fact, actuate on these conditions and provide support for the definition of dynamic safety functions to be performed at runtime.

2.3 Dynamic Risk Assessment

According to Zio E.[248, 251], the main aim of risk assessment is to systematically model the information and knowledge available at the detailed component/basic event level in order to assess the accident risk at the system level. Therefore, classic quantitative risk analysis techniques, as shown in section 2.2, define formal frameworks to model the system risks considering a well-defined set of functions as well as a restricted usage context. In this context, a common framework used to describe the uncertainties in the assessment is based on probability theory,

and particularly on the subjective (Bayesian) theory of probability, as the suitable framework within which expert opinions can be combined with statistical data to provide quantitative measures of the risk [111, 215]. Indeed, the common term used is Probabilistic Risk Assessment (PRA), although Probabilistic Safety Assessment (PSA) and Quantitative Risk Assessment (QRA) are also widely used.

In this context, Dynamic Risk Assessment (DRA) has emerged as a technique that updates the estimation of the risk of a deteriorating system according to the states of its components, as knowledge about them is acquired over time (with time-dependent behavior of the system risk profile) [248, 251]. Note that this definition still considers as a baseline a very clear system definition and brings the concept of risk update based initially on statistical data compared with the current system behavior data[115, 215]. A drawback of using only statistical data is that one must wait until data about accidents or near misses (pre-cursors) becomes available before updating the estimation of the risk indexes. We can also bring together available knowledge from experts about the processes controlled by the system. Moreover, additional information about the degradation mechanisms from the safety barriers failures or context-aware relevant information impacts on the risk estimation. Although the concept of risk assessment takes into account the relevance of the usage context, the DRA definitions and applications only consider subtle runtime degradation of system components. Hence, we define DRA as:

A risk assessment technique capable of updating the estimated risk at runtime considering all aspects relevant to the risk, such as state of the system components, architectural changes, and context elements that interact with the system.

In this sense, DRA has emerged for defining a set of models, methods, and techniques to be performed by dynamic systems at runtime. Overall, it defines executable models at design time and shift parts of risk analysis, estimation, evaluation, and control to runtime. This concept has been applied in fields as diverse as avionics [125], unmanned underwater vehicles [25, 228], automotive [7, 62, 238], and nuclear power plants [36, 123, 210].

Compared with “static” quantitative risk assessment techniques, DRA is capable of dealing with runtime changes provided by novel dynamic systems of systems and cyber-physical systems.

Kurd et al. [124] defined a top-level Goal Structuring Notation (GSN) safety argument for dynamic online risk management in order to assure that adaptive systems can provide structured evidence for certification based on the avionics domain. Although the GSN argument is structured for adaptive systems, we understand that it can be extended for open systems such as SoS. In Figure 4, we present an extension of the GSN safety argument including the requirements for openness as demanded for MCPS as well as risk assessment demands.

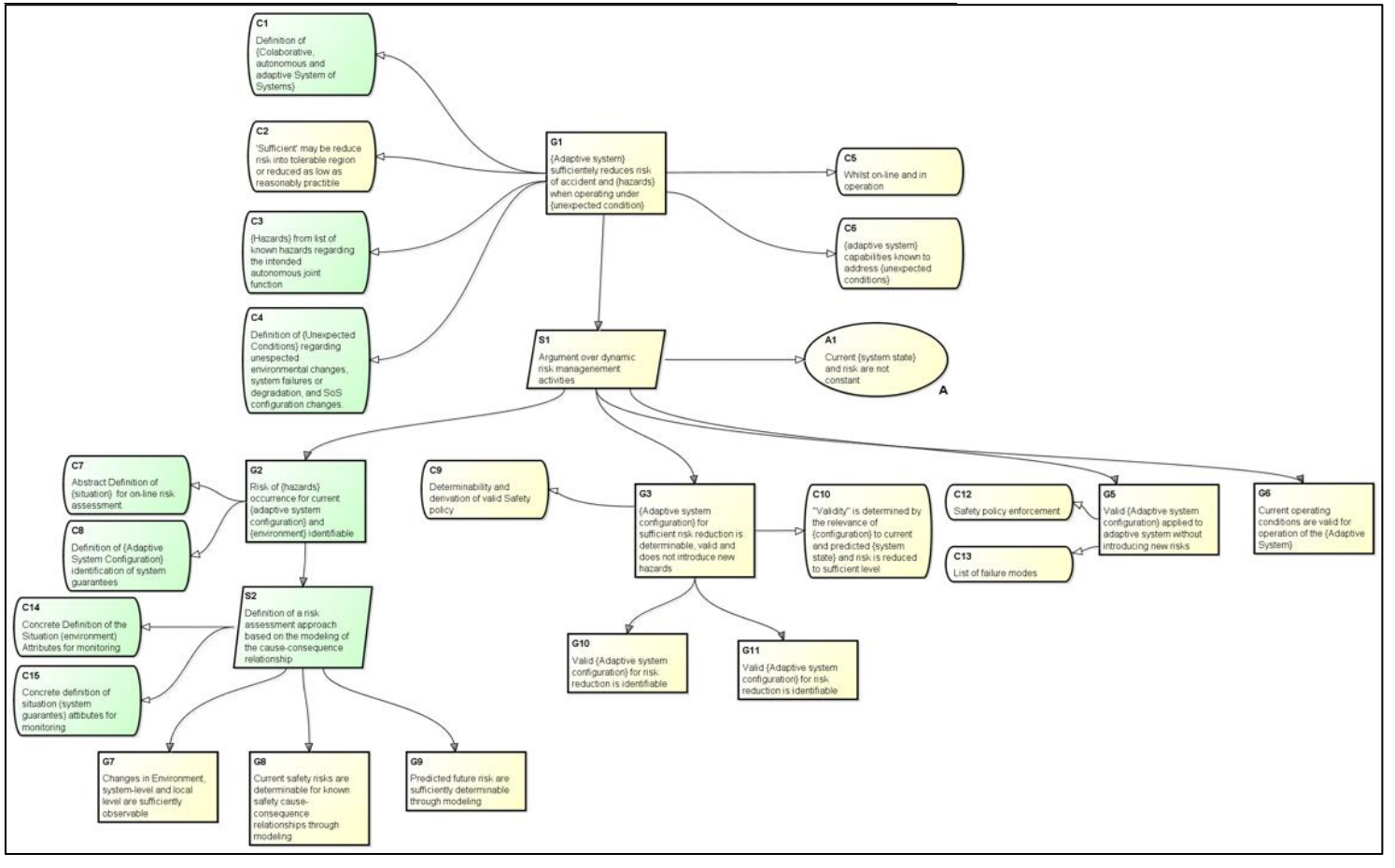


Figure 4 Extended GSN top-level safety argument for dynamic risk management.

The top goal regards the main argument, which comprises the objective of keeping the overall risk in the acceptable region for the system's operational usage. However, we further extended the context assumptions for this goal by considering openness, dynamicity, and architectural SoS changes for the contexts C1, C3, and C4. All in all, it is assumed that the system definition, the hazards list, and potential unexpected situations must be reviewed in order to include new issues and problems inherent in MCPS.

In Figure 4 the argumentation strategy (S1) is based on the main tasks related to risk management, such as identification of all hazards (G2), defined sufficient risk reduction strategies (G3), analysis to check that no further risks were introduced (G4), and validation of the current operating conditions for the system (G5). In this thesis, we focus on the risk assessment that is directly related to the hazard and risk assessment analysis stated in goal G2. At this point, due to the system dynamicity, we consider a basic abstract definition of online risk assessment and a definition of abstract configurations with their respective system guarantees. We therefore refined this goal with a strategy (S2) for defining a risk assessment approach that is based on the concrete definition of a situation based on system and context elements. Finally, we proceed with the relevant aspects for risk assessment as follows: G7 – changes in the environment system level and local level are sufficiently observable; G8 – current safety risks are determinable for known safety cause-consequence relationships through modeling; and G9 – predicted future risks are determinable.

Therefore, this structure determines the overall requirements for complete solutions regarding DRA. It summarizes the concepts and demands from the standards defined so far, and then guides relevant issues that need to be targeted for system development. It is worth noting that the applicability of the framework might vary according to the risk management requirements, the system complexity (dynamicity, autonomy, adaptiveness level), and the demands of standards and accreditation bodies.

In the next subsection, we will provide a detailed analysis of the state of the art in the field of DRA for different domains,

including MCPS. We will describe how such defined DRA issues can vary and be handled in different approaches and domains.

2.4 State of the Art on DRA

As we can observe, DRA has been increasingly adopted by different areas to deal with system dynamicity for safety-critical systems. In this section, we examine the main DRA topics by analyzing different application domains and, at the end of the section, summarize the analysis in a table based on the concepts defined in the previous sections.

DRA for runtime monitoring of Autonomous Underwater Vehicle (AUV) missions

Autonomous Underwater Vehicles (AUVs) must operate in severe environmental conditions that challenge the overall system reliability in terms of the risk of loss. In their concept, "Autonomous" does not mean that no personnel will operate them; rather, autonomy is the system's ability to change its pre-programmed plan of action to achieve its goal [234]. Such systems must consider the vehicle's reliability as a whole, its sub-systems and components, as well as the operating environment to estimate the risk of losing the vehicle. Such systems therefore must perform online monitoring of the risk of loss to decide whether it is worth continuing with the mission.

In the works [25, 227, 228], the authors defined a Bayesian Belief Network for modeling and estimating the risk probability of AUV loss. Their works contribute both qualitative and quantitative aspects of risk models. They defined an informal process to model risk parameters from the experts' requirements and specified a BBN risk model for online estimation of the probability of loss. Hence, they used the BBN for modeling the knowledge of experts instead of pure statistical estimation approaches.

The risk classification is defined according to the probability of losing the AUV. They basically defined a two-region situational space breakdown, namely inadequate or adequate. This establishes what is acceptable or not acceptable according to the concept of the ALARP approach. The authors did not define

further refinements because they did not specify any elaborated risk control actions, besides aborting the mission to avoid the loss of the vehicle.

DRA for Avionics and Autonomous Driving

Adaptive systems do, in fact, exhibit autonomy as another inherent characteristic due to their ability to deal with online decision making to fulfill the final goals. Such systems are specified according to certain levels of autonomy [69, 198, 234] which define the level of independence from human control and supervision. Hence, autonomy and adaptiveness are intrinsically related by their impact on the system behavior and by how they accomplish the main objectives, although there are differences in the behavior specification for the two system attributes [6, 62, 63].

Autonomous systems emerged from the evolution of interaction between several fields of computer science and engineering, such as multiagent systems, artificial intelligence, control theory, and so on [41, 108, 230]. Moreover, over the years, several specific demands for autonomic IT systems have spread over systems in general, such as self-healing, self-configuration, self-adaptation, self-tuning, self-organization, self-stabilization, etc. These requirements have promoted consistent advances of artificial intelligence techniques in order to bring such systems to the next level of adaptation and then achieve autonomy for these specific tasks and purposes. For instance, we have witnessed increased application of autonomous functions in order to avoid accidents in transportation systems, such as airborne collision avoidance system (ACAS/TCAS/ACAS-X) [56, 121], adaptive cruise control (ACC) for vehicles [182, 245], or modern autopilot systems [44, 71] for avionics and vehicles. These functions and systems are built on a complex and adaptive software basis capable of evolving its behavior according to experience and to tune its responses according to each situation.

In spite of all the benefits of autonomous functions, they often struggle with the safety argumentation due to the challenges on the precise specification and assurance of the system behavior [29, 124, 232]. As we have seemed in the previous sections,

Kurds [124] defined a framework for dynamic risk management for intelligent aero-control systems in order to deal with the dynamicity caused by the interaction of the Gas-Turbine Aero Engine control and the environment. The authors argue that some changes in the environmental conditions invalidate all the operational conditions for the considered system, especially for the behavior of control functions in safety-critical software systems bounded to prevent the occurrence of known system level hazards. Hence, a safety monitoring strategy must be defined somehow in order to guarantee that the autonomous or adaptive system will assure that the safety requirements will be fulfilled at runtime despite unpredictable interactions between the environment and the system.

In the literature, safety monitors appear are mainly responsible for realizing risk management techniques [88]. They appear under different names, including: safety kernel [109], autonomous safety system [186], safety bag [119], emergency layer [75], or diverse monitor [91], and so on. In [144], the authors defined an approach for the expression of safety rules based solely on the expertise of analysts. The models are also of great importance for describing the monitor behavior in order to take it into account for the development of the functional layer. In [61], the authors defined a conceptual framework for a safety supervisor on the functional abstraction level to support early design decisions for the development of safety supervisors. In this work, Feth et al. placed the safety supervisor as the key element into the chain of concerns that needs to be regarded in the development of autonomous vehicles. This chain includes safety argumentation, risk reduction strategy, and safety world model, which encompasses situation description, prediction, and risk assessment. Feth et al. argue that autonomy realized with the help of AI techniques can eliminate the need for a precise specification of the system behavior in every possible situation, which makes it practically impossible to analyze such systems.

In this way, the safety community has established that safety supervisors play a key role for the safety assurance of autonomous and adaptive systems, considering the unpredictable system behavior of such systems caused by the complex interactions with the dynamic environment and the system. This concept has also

been adopted by the healthcare community, as we have seen in the ASTM f2714 standard and in the literature. Its application in this field will be presented in the following section.

2.5 Dynamic Risk Assessment for healthcare systems

The healthcare domain has always demanded patient monitoring solutions to enhance treatment, especially for the hospital environment. The range of applications might vary from monitoring a patient's vital signs measured by monitoring devices to modern applications on safety interlocks based on a feedback-control loop. Hence, we classified applications according to their intended functionality, such as diagnostics, condition monitoring, smart alarms, and safety interlocks.

Clinical Decision Support Systems (CDSS)

Clinical Decision Support Systems (CDSS) are computer systems designed to impact clinicians' decision-making about individual patients at the point in time when these decisions are made [18, 20, 187]. Assuming that clinical decision-making is a cognitive and iterative process that deals with uncertainty, such systems emerged to support clinicians in managing an increasing amount of information regarding such procedures. Therefore, clinical decision-making has been aided by a set of analytical tools that process electronic health records to enhance the quality of care and avoid errors during the process.

Metzger et al. [177]defined a CDSS framework for classifying CDSS according to different aspects. However, we highlight the aspect of timing, i.e., when CDSS provide support, which is before, during, or after the clinical decision is made. This determines how active or passive the support is, that is, whether the CDSS actively provides alerts or passively responds to physician input or patient-specific information. Moreover, such systems can model expert knowledge and add historical data processing to estimate a patient's diagnosis (and sometimes simulate human thinking). Hence, when we have an active decision support system processing different patient data to provide or support a diagnosis and thus decision-making [107], it

enables the first aim regarding online patient risk analysis and monitoring, although this was not the intended function initially.

Knowledge-based systems, which are most commonly used, typically organize knowledge about diseases and the associations of symptoms in the form of if-then rules. For example, Dayan et al. [43] introduced traumatic brain injury (TBI) prediction rules in a CDSS to foresee risks of TBI. Laleci et al. [55] utilized a guideline-based CDSS to help manage the personal care plans of elderly people. In [59], the authors reviewed CDSS solutions used for online triage in emergency departments.

Non-knowledge-based systems are usually built upon machine learning techniques that can learn the associations between symptoms and diseases from electronic health records [20, 225]. Compared to knowledge-based systems, these systems greatly reduce the human effort required to manually build and update a large knowledge database [151]. However, these systems typically struggle with lack of interpretability of the analysis results [151] as the dependency of robust datasets to enable the derivation of the effective models [19, 31].

Patient monitoring or sensoring

Patient monitoring refers to the techniques, procedures, and tools that enable continuous patient sensoring in hospitals (wards, ICUs, PACU, etc.) [17, 142, 146, 147, 155, 159]. According to Masimo's CEO "Continuous monitoring allows clinicians to anticipate a serious adverse event at a time that it is preventable by an intervention". Therefore, continuous patient monitoring enhances the quality of care, decreases the risk of adverse events, and supports the hospital operation.

The main problem during a patient's stay in hospital is how to supervisor the cumbersome number of single devices that monitor the patient's vital signs. This problem can grow exponentially depending on the ratio between the number of patients and the number of caregivers. It has been a challenge for hospital administrations worldwide to establish the perfect number of caregivers in their structures. Patient monitoring provides an integrated view of all monitored vital signs, enabling caregivers

to provide better care and actually increasing the awareness level. Patient monitoring has been implemented in hospitals with medical devices such as pulse oximeters, respiration monitors, multiparameter monitors, blood pressure monitors, EEG systems, Holter and Fetal monitors, and so on [17, 142, 146, 147, 155, 159]. Such signals can be either observed separately or analyzed in a combined way using a scale (such as the Pasero, Ramsey, Glasgow coma scale [153]). This can be combined with an alarm system that can be configured according to the thresholds of each vital sign. In this way, as soon as the system detects any threshold violation by the monitored vital signs, it raises an alarm to the caregivers.

In this sense, we have observed increasing adoption of such solutions by the healthcare provider industry. The most prominent monitoring systems are Massimo, Siemens, General Electric, and Philips. However, these commercial systems struggle with conducting an integrated analysis of the vital signs beyond the predefined scales. Therefore, such systems have minimal risk assessment requirements and thus only implement threshold monitoring of the vital signs. Hence, any refined detection or prediction of any adverse event is the burden of the caregivers.

In the literature, we can observe an increasing number of data analysis (vital signs) approaches for risk estimation and situation prediction. Such approaches can be classified according to the inherent methods and models as either top-down or bottom-up. The former are the natural trend of the classic methods and models used for CDSS. Top-down approaches aim to elicit experts' knowledge and implement physiological models for risk estimation to incorporate the time history of multiple measurements into the MCPS. In this field, there are several works, such as [97], where the authors propose a monitor for blood oxygen concentration that predicts critical drops in oxygen levels caused by pulmonary shunts in infants. Here, a statistical approach is used to handle inaccurate models and sparse data is the parameter-invariant detector [241]. In [105], the authors propose a self-adaptive evolutionary sepsis screening system to shorten the time for syndrome detection and improve the positive effect of treatment. Here, the screening frequency and content can be automatically adjusted according to the current status of the

patient. The approach is based on Dynamic Bayesian Networks (DBN) to estimate the onset probability or mortality of the elderly patient crossing some predefined thresholds. In [13, 173], the authors established formal timed automata models for safety property analyses of closed-loop scenarios for patient-controlled analgesia. Although Pajic et al. formalized the scenario and contributed the first safety analysis for MCPS actuation scenarios, the authors focused on the safety properties defined by the user's vital signs and did not consider other relevant system data (e.g., system configuration changes), context data (e.g., drug type), and patient data (e.g., patient history details).

Top-down approaches have contributed the first implementations of risk monitoring strategies for CDSS, Smart Alarms, and MCPS. However, they might struggle with the unpredictability of physiological human processes as well as limitations of experts' knowledge and models. Recently, the explosion of data exposition has promoted a propitious environment for data-driven bottom-up approaches. Such approaches can learn parameters and structure models based on data. Moreover, advances in AI/machine learning techniques such as neural networks and Big Data Analytics have improved the accuracy of risk models and situation prediction. However, there are often practical challenges to data-driven approaches in the medical domain, namely, algorithms require rich patient data representing the entire population with accurate annotations [11, 22, 73]. Satisfying this constraint is especially difficult in the surgical settings considered in this work where accurate annotations are not a primary concern of clinicians [49, 213]. Thus, it is currently unlikely that a purely data-driven approach will perform as well as top-down techniques in the areas of critical pulmonary shunt prediction, PCA closed-loops, or cardiac monitoring, since data is scarce and frequently missing, and the dynamic response is affected by a high number of potential variables [11, 45, 185].

Besides, we have found data-driven approaches used for detection of anomalies in the isolated vital signs such as ECG, which training datasets are very common to find out [104, 237]. We also have observed DBN being used in monitoring heart rhythm based on ECG data [246], with the main purpose of the system being to identify arrhythmias, which constitute a complex pattern

recognition problem. Moreover, patient monitoring systems have even evolved into the Internet of Medical Things (IoMT), which represents the advent of pervasive, wearable, and implantable sensors connected through a network of smart devices and ambient living sensors [70, 94, 190, 239]. This evolution has brought more accurate analysis, increased the reliability and availability of the monitoring, and enabled a variety of new applications.

Smart Alarms

Alarm systems are part of the basic risk control measures for medical devices, as specified in the standard IEC 60601-1-8[92]. Alarms encompass a set of audible and visual signals that communicate some information from medical devices to the operator considering the process controls by the medical device. Normally, such alarms are classified according to the criticality/priority level of the signal. According to [235], such alarms can be classified as patient alarms, which focus on the health and wellbeing of the patient, and functional alarms, which monitor the operation of the interoperability setup. According to [236], there may even be safety alarms that recommend some actuation in order to guide caregivers and positively change the patient's status.

Every single medical device must fulfill its own requirements for alarms. This often results in alarm flooding situations, which are common when there are several devices in a ward or ICU in a hospital environment [49, 138, 158, 226]. In the literature [50, 172, 207, 218], alarm flooding and alarm fatigue are common hazardous situations for different types of treatments.

In this sense, the standard IEC 60601-1-8 defines the concept of Smart Alarm systems, which aim to aggregate all the alarms raised by different devices in order to decrease the nuisance level when several medical devices are operating at the same time. Such systems monitor different vital signs (provided by the medical devices) and evaluate whether an alarm must be raised. This evaluation is done through a risk metric implemented by top-down approaches such as [98, 137, 173, 241] or bottom-up data-

intensive technique such as the Fuzzy Expert System [45, 138] or machine learning techniques [84].

In the works of Ivanov et al. [98, 99, 241], an approach for CMPS monitoring based on parameter-invariant (PAIN) was developed. PAIN monitors are designed such that unknown events and system variability minimally affect the monitor performance. The solution therefore aims to model the uncertainty of the patient physiological model in order to enhance the accuracy and precision of system alarms raised to caregivers. The authors validated the approach through the detection of critical pulmonary shunts in infants and the related vital signs for predicting such events. Their models are based on predefined vital signs and their respective modeled behavior instead of taking into account the whole associated uncertainty. In [98], the author developed a model for monitoring not only the physiological model but also some context-relevant variables from the clinical environment provided by caregivers. However, such models are still founded on a clear set of parameters and are not specified by a risk assessment model. Moreover, the authors considered the system dynamicity in terms of nuisance from the devices, while system adaptation and changes in the configuration are not targeted by the risk metric. Furthermore, context is barely considered in their risk metric, and more details could be considered in the risk metric, such as the patient's history, treatment details, and so on.

Safety Interlocks

Finally, safety interlocks have been proposed as feedback-control loops for monitoring patient vital signs and actuating on the care delivery devices to avoid hazardous situations for the patient, such as overdosage [13, 72, 173]. The standard ASTM F2761 defines a set of relevant clinical scenarios and application contexts for MCPS-relevant actuations aimed at avoiding accidents resulting from the complex interaction between medical devices in hospitals. These definitions encompass, for example, scenarios such as patient-controlled analgesia (safety interlocks) and physiological closed loops for maintaining blood glucose concentration within a normal range. Both scenarios define a set of monitoring devices connected to the patient and their

respective connections with a centralized system that continuously gathers and analyzes data and, according to the patient's status, sends control signals to the actuation devices.

Currently, due to challenges related to safety assurance and certification, interconnected closed-loop control is still an ongoing field of research for most of the healthcare industry and most medical devices manufacturers [35, 173]. The main issue is reasoning about patient safety in closed-loop scenarios. MCPS are built upon the interconnection of medical devices that autonomously make decisions based on their patient's model. Therefore, we must deal with complexity in order to manage a reliable network of certified device interconnections (MDPnP.org/HL7/ establishes some guidelines for dealing with communication and interoperability). Furthermore, the dynamics of the patient's body are not well understood and exhibit parametric uncertainty during medical treatment. Moreover, a particular challenge arises from the complexity of the interplay between the continuous dynamics of the patient's reaction to treatment and the discrete nature of the controller and communication network. Hence, several works have contributed different challenges provided by such endeavors.

Pajic et al. [13, 117, 173] defined a simulation model that allows us to prove the safety properties of devices on the modeling level and ensures that abstract models used in the verification process are sound with respect to the actual dynamics of the system. This approach deals with the uncertainty exhibited by the interplay between the continuous pharmacokinetic dynamics of drug absorption by the patient and the discrete model of the controller and communication network. A procedure was developed that utilizes convex optimization to calculate an upper bound on critical timing values for a linear time- invariant (LTI) model with uncertain parameters and shows that the behavior of the UPPAAL model is always within this bound. The authors evaluated the results by analyzing a clinical scenario with pulse oximeter sensors and an infusion pump for PCA. However, the authors considered a well-defined clinical scenario for validating their experiments. In this context, establishing safety properties is not a challenge and the key aspect focuses on timing aspects and the acceptable thresholds for the properties. Moreover, the authors

did not consider the issues raised by the system dynamicity (autonomous adaptations or multiple configurations), which can undermine all the assumptions made for safety property verification. Finally, the work also lacks some risk assessment to deal with compliance with standards.

In [137], the authors present the results of efforts aimed at enabling context awareness for a special type of safety-critical system. They propose enriching MCPS with various non-medical devices, e.g., RFID tags/readers and video cameras, that cooperate to capture various implicit context information during a medical procedure to reason about human errors. In this sense, human errors can be detected at early stages with much higher probability, and thus, the MCPS can trigger corresponding actions to reduce/prevent the potential risk of human error. A very relevant aspect of this work is the introduction of a safety assessment that evaluates the risk of mitigation actions (triggered by context-aware evaluations) based on solid medical knowledge. For this purpose, the authors developed a context uncertainty model that captures the MCPS dynamicity and their context together. However, although the sensoring uncertainty is considered as an element of the model, the authors did not consider the overall uncertainty exhibited by multiple configurations. In the study, the authors only considered a fixed system architecture and therefore no potential adaptation or reconfiguration. Hence, the uncertainty model lacks the system dynamicity information. Moreover, the authors assumed specific details about the clinical scenario and did not delve further into risk identification parameters to identify which parameters should be considered the most relevant for the risk model.

2.6 Overview and classification

		Adaptiveness			Risk Assessment			
Category	**Approach**	**Reconfiguration Technique**	**Adaptation Strategy**	**Autonomy Level**	**Identification**	**Analysis**	**Classification**	**Control**
Standards	**ISO 14971**	not specified/ recertification required	process-oriented	not specified	required by the process	required by the process	required by the process	required by the process
	IEC/ISO 62304	not specified/ recertification required	process-oriented	not specified	required by the process	required by the process	required by the process	required by the process
	ASTM F2711	Parameter & Structure	pre-engineered variability	supervised autonomy of the safety monitor actuation	not specified	not specified	required by the process	not specified
	IEC 60601-1-8	not specified/ recertification required	process-oriented	not specified	specified	specified	specified	recommended
CDSS	**Care-pre**	not predicted	not predicted	not predicted	defined by the reasoning model	Deep Learning technique for	defined by the reasoning model	not specified

		Adaptiveness			Risk Assessment			
Category	**Approach**	**Reconfiguration Technique**	**Adaptation Strategy**	**Autonomy Level**	**Identification**	**Analysis**	**Classification**	**Control**
						prediction		
	Dervishi's Fuzzy-stratification	not predicted	not predicted	not predicted	defined by the reasoning model	Deep Learning technique for prediction	defined by the reasoning model	not specified
Smart Alarms	**Statistical Parameter-Invariant (PAIN)**	parameter	pre-engineered variability	not predicted	pre-defined	PAIN statistic	PAIN statistic	alarms/ human in the loop
	Smart Alarms paper	not predicted	not predicted	not predicted	defined by the reasoning model	defined by the reasoning model	defined by the reasoning model	alarms/ human in the loop
Patient Monitoring	**Multifusion BBN**	not predicted	not predicted	not predicted	pre-defined	BBN predefined	BBN predefined	not specified
	Auto-BN	not predicted	not predicted	not predicted	pre-defined	Adapt-DBN	DBN predefined	not specified

		Adaptiveness			Risk Assessment			
Category	**Approach**	**Reconfiguration Technique**	**Adaptation Strategy**	**Autonomy Level**	**Identification**	**Analysis**	**Classification**	**Control**
Safety Interlocks	**Pajic & Arney approach**	not predicted	not predicted	autonomous actions for safety interlocks	pre-defined	timed-automata Simulink model	pharma-cokinetic patient model	safety interlocks
	Context-aware PCA-MCPS	not predicted	not predicted	autonomous actions for safety interlocks	pre-defined / context-aware	BBN predefined	BBN predefined	safety interlocks
	Presented approach	**Parameter & Structure**	**Goal-based selection**	**Autonomous actions for safety interlocks**	**Model-based approach for risk parameters elicitation / context-aware**	**Adapt-BN**	**Specified by the approach**	**Safety interlocks**

2.7 Summary and conclusion

2.7.1 Unpredictability of MCPS configurations and guarantees

As we have seen for the whole chapter, MCPS's inherent characteristics such as autonomy, system dynamicity, and openness have challenged traditional safety assurance techniques, which complicates certification and thus hinders easy and wide-spread utilization of such systems. Recent solution ideas such as runtime safety certification (e.g., Conditional Safety Certificates – ConSerts [201, 203]) tackle the challenges of system dynamicity and openness by shifting certain safety checks from development time to runtime. ConSerts are based on predefined guarantees and demands, which are formalized assumptions regarding each constituent system's environment. These assumptions/demands can be evaluated dynamically and, eventually, top-level guarantees can be determined for a dynamically formed system composition.

Once the top-level guarantees have been determined, the question arises whether these are actually sufficient for the current system context. A PCA treatment setup, for instance, could be utilized in different clinical scenarios, each implying specific top-level safety requirements that need to match with the current system guarantees. Now we could either go with a worst-case assumption, i.e., take the most critical PCA scenario and derive the requirements from that. This would lead to many situations where the current top-level guarantee would not be deemed sufficient and hence certain constraints would be required (e.g., not allowing the application to run at all, only allowing a degraded mode, or demanding the presence of a human in the loop).

Although several works have advanced the field of runtime safety certification advances, we also need to consider the impact on the risk assessment of system adaptations and context. As can be seen in the overview table in Section 2.6, most approaches barely consider adaptiveness as a key element of the solution. With regard to adaptiveness aspects, only the ASTM F2711 standard and the PAIN approach accept very limited adaptiveness. Therefore, we can observe a lack of approaches that take into account such aspects for dynamic risk assessment.

2.7.2 Limited risk models for dealing with the dynamicity of MCPS

An accurate risk model plays a fundamental role for enabling runtime risk assessment of MCPS. The performance of the current risk runtime assessment is very relevant for the widespread adoption of MCPS. A fundamental problem leading to increased numbers of false positives and false negatives is the fact that alarms (as well as other triggers) are typically based on fixed thresholds of one or a few monitored parameters. Given the very volatile nature of the relevant context conditions (e.g., the patient's condition, which is characterized by a potentially complex interplay of different vital signs), fixed thresholds are a weak means for assessing the current risk.

As can be seen in the table, risk identification must enhance the completeness of the risk parameters in order to get an accurate and efficient risk model. Most of the approaches consider a fixed set of parameters and their respective thresholds based on the medical literature. Such risk metrics (MEWS, NEWS, EWS, etc.) were defined for patient monitoring by caregivers; the first assumption therefore is that either risk assessment or risk control should be performed by the caregivers. Even though such risk parameters are relevant, however, some works [38, 84, 142, 147, 155, 161] have highlighted the relevance of other elements related to these first-order parameters as well as issues directly related to the behavior of such parameters. Deep Learning could be used for learning and adjusting relevant parameters; however, data-intensive strategies are limited by the available data sources, which only gather data about the patient (and not data about the system states). Hence, increasing the completeness of the risk models might be a key factor for enabling dynamic risk assessment.

2.7.3 Complying with best practices described in recommendations by standards

The ISO 14971 standard clearly demands a formal plan for risk management that comprises analysis, assessment, evaluation, and more. According to the standard, "*For each identified hazardous situation, the manufacturer shall decide, using the criteria defined in the risk management plan, if risk reduction is*

required". Therefore, the manufacturer needs to somehow assess and evaluate whether the current situational risk level is acceptable for the current function.

Although the standards do not specify any concrete way for implementing a risk evaluation plan or an acceptance level, they recommend analyzing similar projects, overall regulations, clinical data, the state of the art, and best practices to come up with a concept of the risk and its acceptability for the specified function.

Furthermore, we have found in the literature the definition of appropriate tasks for risk evaluation, such as the ISO 14971 standard, which establishes risk evaluation as *the process of comparing the estimated risk against given risk criteria to determine the acceptance*. Nevertheless, this definition is fully focused on the system design phase of the product's lifecycle. Once the system is in operation, the standard does not predict (or even allow) any structural changes in its design. All the countermeasures are therefore specified and performed at design time.

Moreover, not only the context is dynamic but also the system itself is capable of changing its constituent parts and providing different levels of nominal functions or even emergent behavior. This possibility is not predicted and sometimes even forbidden by the standards. Therefore, in the case of changes in the system, its operational usage must be suspended and if the system is to be used again, it must be recertified according to the new design, which is impractical for both care providers in hospitals and manufacturers [223].

2.7.4 The need for dynamic risk models for enabling runtime risk assessment of MCPS

Alternatively, MCPS could become context- and risk-aware, meaning they could be enabled to perceive their context and dynamically determine the current risk and corresponding safety requirements of an MCPS application such as patient-controlled analgesia (PCA).

Moreover, we understand that an effective risk model acts as a key driver for enabling dynamic risk assessment. Such a holistic model should bring together the complexity of techniques dealing with the openness and adaptiveness of MCPS and risk assessment approaches for enabling dynamic risk assessment for such systems.

Runtime risk assessment methods have been considered as a potential solution for the stated problem. Such techniques are capable of updating the estimated risk of a deteriorating process according to the performance of the control system and its context factors. Although such techniques have advanced the state of the art for autonomy issues, they still struggle with dynamic composition. Moreover, the risk metrics derivation approaches have been limited due to the incompleteness of the risk parameters and the fuzzy definition of situational spaces for establishing safety monitoring.

With that said, we state the core problem addressed in this thesis as follows:

> The lack of novel dynamic risk assessment approaches is a key reason for the inefficacy of the current risk metrics and consequently imposes severe limitations for the safety assurance and performance of CPSoS.

3 Solution Overview

The main contribution of this thesis is to enable dynamic risk assessment for cooperative medical cyber-physical systems. It provides an approach for engineers to derive dynamic risk analysis artifacts and establish runtime risk evaluation models considering all dynamic context requirements for the situation and the effects of the system configurations on the risk metrics. Therefore, the main achievement is to provide a fine-grained optimization of a system's safety performance and increase system availability for open and adaptive cyber-physical medical systems.

Risk assessment defines a technical framework for the structured analysis of the risk associated with the system behavior. Hence, the main task is to organize the information and knowledge available at the detailed component/basic event level in order to assess the accident risk at the system level [37, 91, 95, 170]. The ISO 31000 standard assumes that the context and the system (item) are defined for the execution of the main processes that risk assessment encompasses: risk identification, risk analysis, and risk evaluation (see Figure 5, the dashed squares highlight the focus of the contributions). Risk identification deals with finding, recognizing, and describing risks by defining risk sources, hazardous situations, and potential consequences. Risk analysis is used to comprehend the nature of the risk and to determine risk levels through quantitative and qualitative analysis. Risk evaluation takes the inputs from the risk analysis and establishes risk criteria for defining the acceptance level for the current risk. Based on the output of the risk evaluation, mitigation actions are specified for keeping the risk within the accepted parameters.

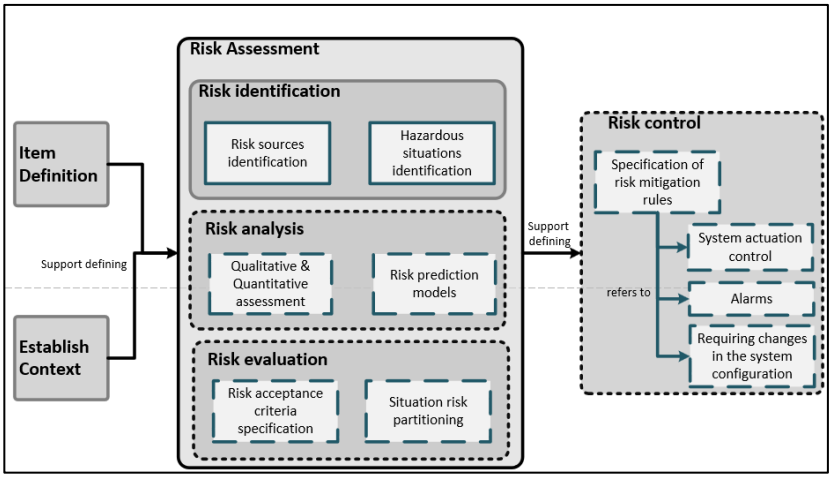


Figure 5: Risk assessment and evaluation: general tasks

As we have discussed in the previous chapters, risk assessment approaches require the identification of all static and predictable contexts/systems for developing the risk analysis and evaluation tasks. Then, the whole risk assessment, evaluation and mitigation process are performed before the system's deployment and operation. Therefore, even though some minimal dynamic aspects can be considered for the definition of the safety concept at design time, they often miss out relevant time-dependent variations that occur as components and systems operate, age, fail, are repaired and changed [110, 131, 231, 248]. Compared to traditional "static" quantitative risk assessment, DRA is, however, capable of capturing the time-dependent behavior of risk indexes and providing a more realistic description of the system risk. Traditionally, DRA methods used only statistical failure data and a limited set of context variables to monitor a system at runtime. Moreover, such methods struggle with the combined analysis of system and context variables. These challenges have led to insufficient risk awareness caused by fluctuating context conditions, insufficient context awareness, and a lack of reasoning capabilities to deduce the current risk for dynamic systems, such as cooperative medical cyber-physical systems.

A fundamental aspect related to the insufficient risk awareness of DRA is the lack of methods for deriving a set of all parameters relevant for the risk analysis. Lack of completeness of the risk parameters might lead to inaccurate risk metric derivation.

Therefore, this low quality of context awareness prevents a system from precisely assessing the current situation in order to update the risk. In the healthcare industry, for example, several accidents (wrong dosage, alarm flooding, etc.) are reported every year due to the lack of awareness of the systems [40, 142, 207].

Besides the lack of context awareness, DRA methods struggle with considering the impact of system changes on the risk metric. State-of-the-art approaches for condition-monitoring-based risk assessment consider data about the individual degradation process of the target system and of the safety barriers to enable updating the reliability values before actual failures occur. However, we still need approaches for enhancing the accuracy of risk metrics at runtime. For example, cooperative cyber-physical systems change their configuration over time, and this dynamicity might also affect the overall system behavior. Moreover, such systems provide autonomous functions (such as safety interlocks, closed loops, etc.) and strongly depend on an accurate risk model for situation modeling and/or risk countermeasures. Hence, the impact of system dynamicity must be considered for the whole risk assessment process.

To overcome these problems, this thesis aims to provide a set of methods for improving the risk awareness of systems and equipping them with the capability to dynamically calculate the risk of the current situation. In the first contribution, we introduce a corresponding runtime risk analysis approach based on an executable risk model. The risk model is specified as a Bayesian Network (BN) according to a methodology that is also presented in this thesis. The methodology utilizes a metamodel to enhance the identification of relevant risk parameters according to domain-specific information, system dynamicity, and safety engineering tasks. Hence, this work enhances the state of the art by dealing with the lack of completeness of risk parameters and their integrated analysis in a risk metric.

Furthermore, we specify an adaptive risk management model capable of dealing with system configuration changes at runtime. To this end, we define adaptive risk models so that risk monitors can properly respond to system and context changes.

Figure 6 depicts how the contributions are organized to address the challenges identified in the state of the art. To enable a fully dynamic risk assessment for cooperative medical cyber-physical systems, it is necessary to provide an adaptive risk metric that comprises all relevant risk parameters and reflects the impact of system dynamicity on runtime risk assessment. Moreover, the risk evaluation needs to consider a dynamic risk assessment model and a novel situation partitioning model for managing risk metrics and activating proper risk countermeasures based on the current system configuration.

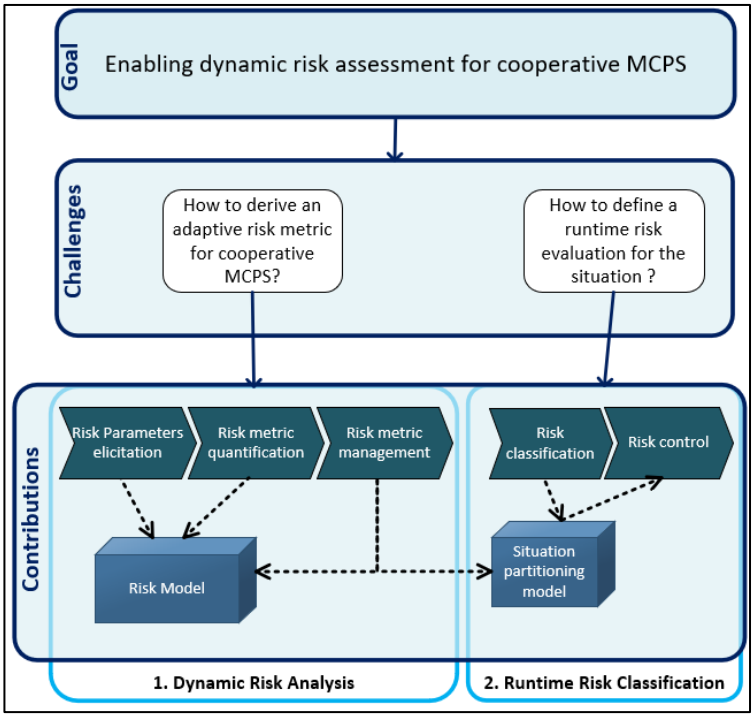


Figure 6 Risk management framework for cooperative medical CPS/SoS. Approach overview with the constituent approaches allocated to the contributions

Within the scope of this thesis, we defined the following two main contributions:

3.1 Adaptive risk analysis for DRA of cooperative medical CPS

This thesis establishes a model-based approach for conducting risk analysis and risk evaluation tasks in order to enable dynamic

risk assessment for cooperative medical CPS. Such a technique needs to specify a set of artifacts and models at design time in order to perform assessment and evaluation at runtime. Hence, the key element to support this task is a meaningful risk metric. In the autonomous automotive domain, for instance, a fundamental issue is to find an appropriate metric for determining the risk due to "collisions", as this is the major hazard of concern. In the medical domain, on the other hand, it is mandatory to consider a huge variety of therapy-specific procedures for a patient and their complex interactions with the context. For example, some procedures such as radiation therapy can harm personnel as well. In this sense, we provide models and guidelines to support the derivation of a Bayesian Network-based risk metric in order to enable qualitative and quantitative assessment at runtime.

For the risk analysis approach for DRA of cooperative medical CPS, we established three phases: (a) risk parameter elicitation; (b) risk metric specification; and (c) risk metric management.

Risk parameter elicitation

To derive a risk metric for runtime risk assessment, the first challenge is the elicitation of all relevant parameters. For the sake of completeness and efficacy, it is necessary to deal with multidisciplinary and (sometimes) conflicting concerns from different stakeholders. Furthermore, this complex information needs to be cataloged and structured considering key characteristics regarding adaptiveness, openness, and safety criticality, and their inherent capacity to change the environment.

In Figure 7, we present an overview of the elicitation approach. We assume that artifacts produced in the concept elaboration phase (such as system definition models, context and scope definition, intended functions, interactions, etc.) and models from the risk identification phase (such as hazard analysis models, foreseeable situations, failure modes, hazardous situations, cause consequences analysis, and so on) should be evaluated to elicit the parameters and develop the risk assessment model. Other runtime certification models are also highly relevant, but not mandatory. Runtime certification techniques provide useful

formal models and languages to specify such information, such as defined safety properties, requirements analyzed for quality attributes that can be derived from the analysis of hazard and system definition models.

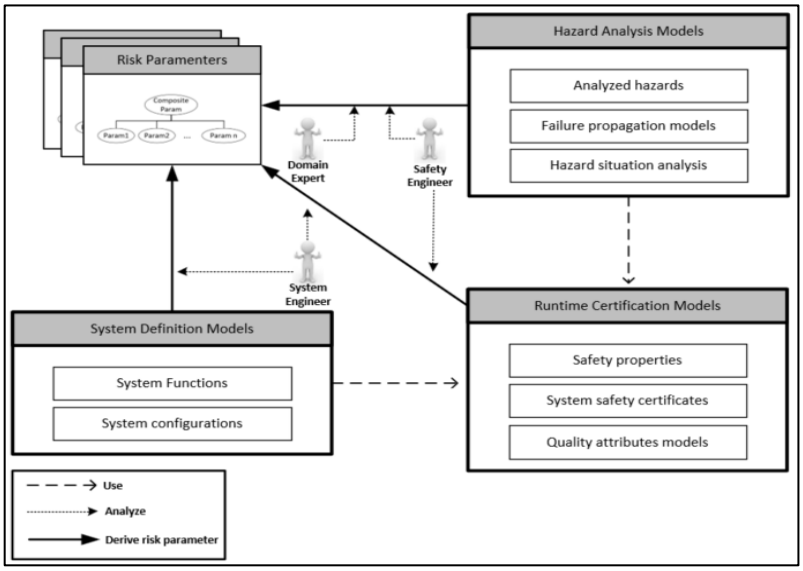


Figure 7 Risk parameter elicitation

We also provide a meta-model for structuring the relevant information considering domain-specific elements, risk assessment, system details, and safety guarantees to build the assessment model. Such a model needs to specify all classes for the relevant information so that domain experts as well as safety and system engineers can derive a minimal set of relevant information according to the main standards and guidelines of the domain.

Risk metric specification

Once the risk parameters are identified and classified, it is necessary to build a risk model that establishes quantitative values for the risk parameters according to their relevance for the current situation. We use as a foundation the Bayesian Belief Network, due to its inherent tree-structured model, which is capable of dealing with probability and uncertainties. Furthermore, elements related to safety guarantees and adaptation between different

certified configurations are considered to develop an accurate risk assessment model. The proposed framework defines a technique for specifying an extended risk assessment model and its transition specification according to the current system configuration.

In Figure 8, we show an example of how the parameters can be structured to build a risk metric. The decomposition framework is responsible for providing a classification of the parameters. Based on that, it deploys the referenced parameter in the risk metric (tree-structured). Moreover, the built structure provides a framework for the quantification of each parameter and its respective aggregation rules according to the risk assessment. The metamodel defines classes of parameters related to risk assessment, system properties, domain-specific elements, and safety guarantees. In the next chapters, we will specify in depth all elements, their relationships, and how they can be used to derive a risk metric for runtime risk assessment in the context of cooperative cyber-physical systems.

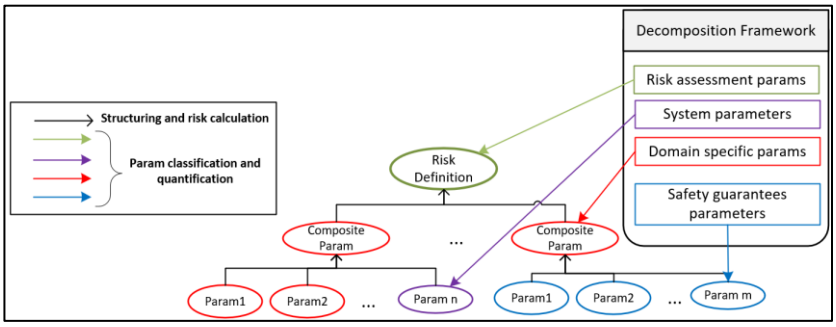


Figure 8 Risk metric derivation

Runtime risk metric management

Once a risk metric has been defined for the situational assessment, the proposed approach defines an abstract analysis of the system configurations to define the risk metric management. We take into account that adaptations might be either performed by the autonomous decision of the system or be triggered by an external entity, such as monitors, managers, or even a human administrator. Such adaptations have an impact on the systems involved in the current cooperation and thus affect the elements

responsible for performing some steps of the risk metric. Moreover, the identification of all possible configurations might be impractical and cumbersome due the potentially large number of combinations of systems and devices. Hence, we provide a conceptual model for risk metric management based on abstract configurations, which might vary according to the elements of the risk metric. This makes it possible to define risk metrics based on services, while the transition between them is defined based on the adaptations.

In Figure 9, we can observe an example of risk metric management for three potential abstract configurations. Each abstract configuration comprises a set of services provided by cooperating systems that implements part of the current risk metric. However, when a relevant adaptation occurs, it changes the provided services in the cooperation, so it must trigger a change for the respective risk metric for the current configuration.

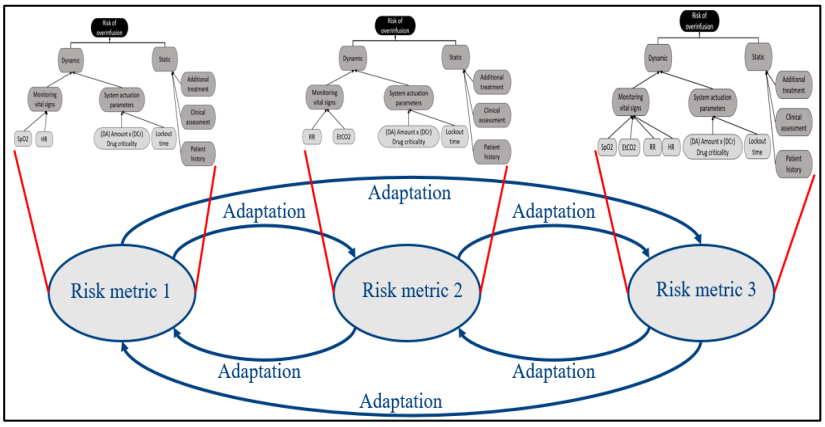


Figure 9 Risk metric management based on abstract system cooperation

3.2 Runtime risk classification and risk control specification

We provide an extended situation analysis framework to support the definition of situation-partitioning states. Regarding the situation risk, we provide a classification structure for the risk level considering recommendations found in standards on accident severity and probability. As part of this approach, we provide a formal situation definition and a class-based classification, which varies in terms of the precision and accuracy

to clearly identify accident conditions as well as the sensitivity to recognize a set of hazardous situations that are not harmful yet. This situation-partitioning framework is a fundamental tool for the specification of proper mitigation actions, which will be explained in the next sections.

Once we enable the system to analyze and monitor the risk of the current system configuration and context, risk evaluation tasks and countermeasures need to be defined considering all potential situations. We assume that each partition imposes new requirements on the system's behavior. For example, in Figure 10, risk metrics with a lower integrity level (IL) are suitable for situations with lower risk levels, such as S1 in the figure. In these cases, accuracy and precision of accident identification is not required as much as sensitiveness (ability to identify situations on the brink of becoming dangerous). For these situations, configurations providing these characteristics must be enabled. On the other hand, situations classified as high-risk level need to have a system configuration that provides a higher integrity level regarding accuracy and precision in order to identify an accident (respiration distress in the example presented in Figure 10).

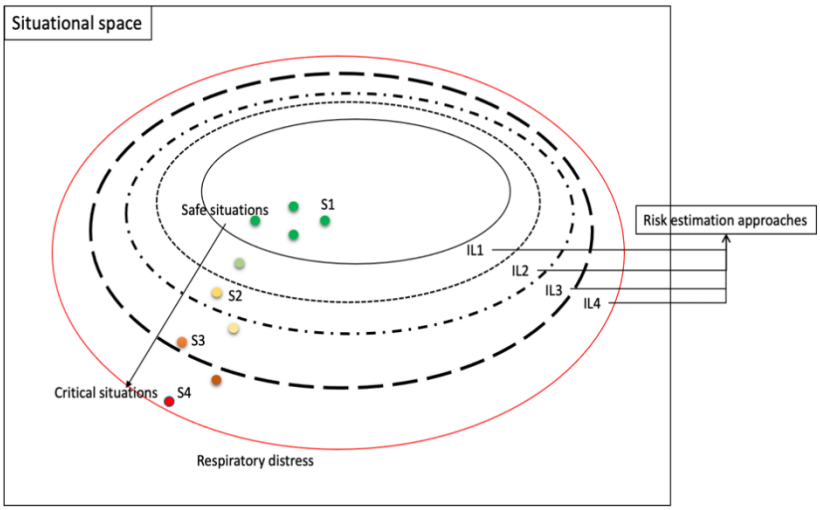


Figure 10: Situation partitioning and risk estimation approaches

To specify risk control measures, the approach considers the current risk level and evaluates the minimal safety guarantees that are suitable for the current situation. Runtime safety certification

approaches provide a formal language for specifying the safety guarantees for system configurations. Such guarantees are not only utilized in the risk assessment model but also in the specification of risk measures. Hence, the proposed specification of risk control measures considers alarms and warnings to alert the system environment about the risk level and recommended actions; mechanisms for bringing the system to the safe state; triggering adaptation mechanisms for deriving a new configuration; and a certification process for achieving new guarantees according to the given configuration.

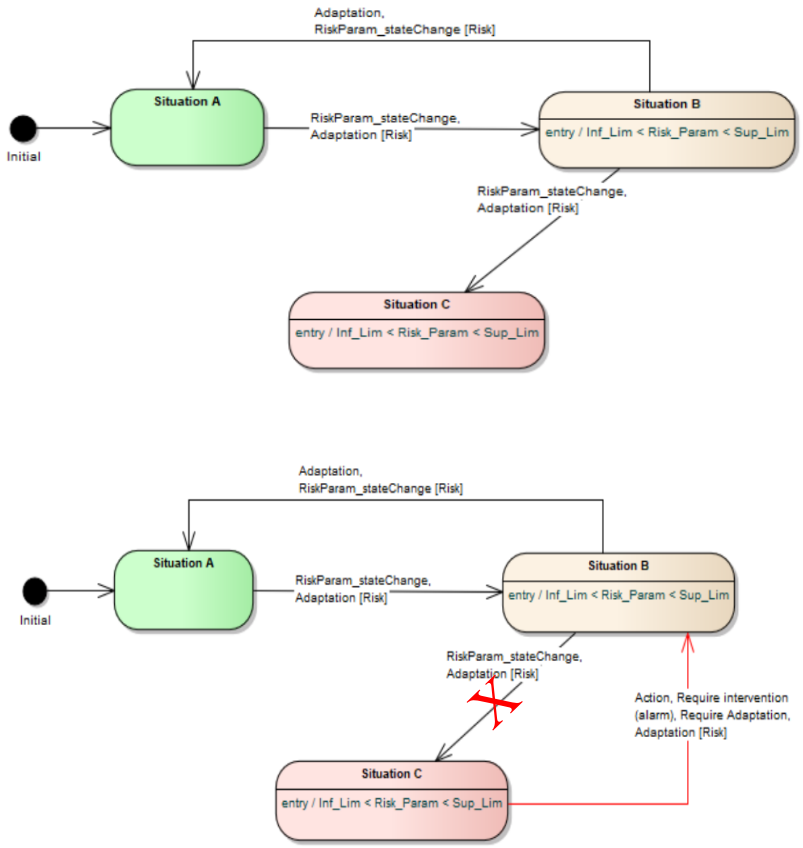


Figure 11: Specification of risk control measures through the inhibition of the reachability of critical situations.

In Figure 11, we present an example of the generic framework utilized in the approach. During the risk assessment phase, the situation-partitioning structure is defined. In Figure 11, the

situation's criticality is represented by the color of the situation: Situation A (green) is a safe situation, situation B (yellow) exhibits a higher risk level, and situation C (red) is a critical situation. Next, the analysts need to identify all the transitions that might lead to the set of critical situations and apply one of the risk control actions defined by the framework to inhibit the transition. In the example depicted in Figure 11 (note the red X), all the potential adaptations or system behaviors that might lead any risk parameter to a critical situation should be avoided. Therefore, every single action, alarm, or adaptation must be specified (note the red arrow in Figure 11) in order to keep the risk parameters within the predefined acceptance levels at runtime.

The main aim of the approach is to improve system performance through proper system actions according to the current risk level. The aim of the risk assessment contribution is to identify the current situational risk and specify the actuation of risk control measures in the system to manage the assessed risk. Both risk assessment and risk control action utilize safety certification information to assess the current risk and define new configuration requirements at runtime. Therefore, we enable situation analysis at runtime in order to enhance safety certification approaches in terms of the evaluation of suitable configurations. Furthermore, this approach enhances the safety performance through dynamic risk monitoring of the system and specification of the countermeasures which might enable (or require) proper system configuration according to the current risk.

The risk assessment and the control technique consider the risk evaluation and the available risk control actions to improve the system availability through actions related to system adaptation and behavior. Hence, this strategy defines the required configuration's integrity level for the current situation risk demands. Moreover, it specifies different levels of system actuation depending on the current situation, constraining or enabling system behavior according to the current risk level and the current configuration integrity level.

We envision as the main contribution of this thesis that it enables systems to reason about safety regarding the overall situation and then empowers systems with decision-making capability to

activate a suitable configuration according to the current risk. System functions can hence be provided in scenarios where they could not have been provided with a conservative and limited analysis performed at design time.

4 Adaptive Risk Analysis for DRA of Cooperative Medical CPS

This chapter describes the contribution adaptive risk analysis for dynamic risk assessment of cooperative MCPS. This contribution is established on two constituents that are required for the operationalization of the risk parameter analysis: (i) the risk model for parameter identification and elicitation; and (ii) a guideline for building risk metrics for runtime risk assessment. Such constituents together provide ground basis for the risk metric management, as you can see in the Figure 12. These constituents are presented using mathematical descriptions, meta-models, and SysML models.

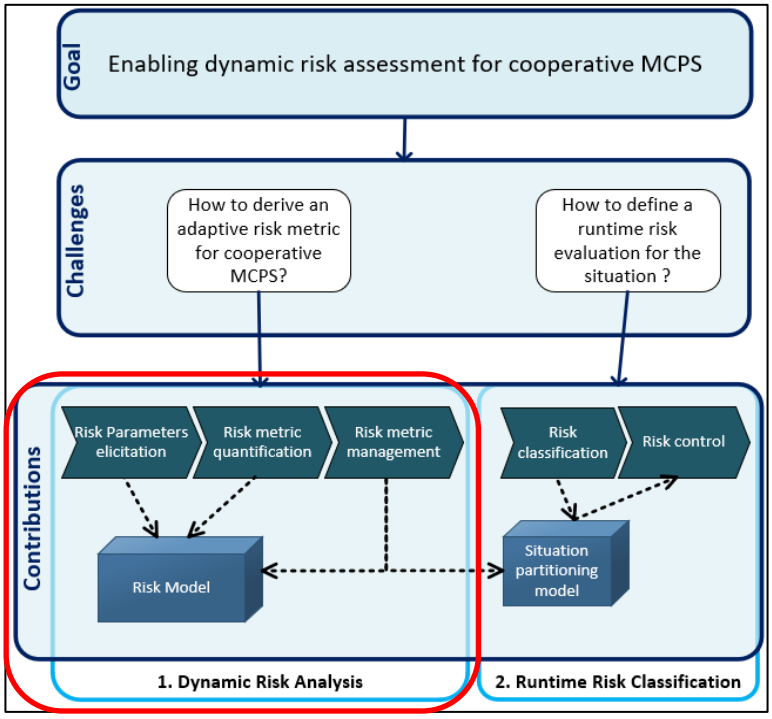


Figure 12 Risk management framework for cooperative medical CPS/SoS. Focus on the Dynamic risk analysis contribution

A running example introduced in Section 4.1 will be used to demonstrate the different aspects of the operationalization based on a real-world example.

4.1 Running Example

4.1.1 Medical context

Effective pain management is essential for patient satisfaction, quality of care, and institutional compliance with international standards and worldwide institutional protocols [147]. Patient-controlled analgesia (PCA) is a widely used, effective method of opioid administration for postoperative pain management. In hospital wards, this treatment consists of a needle attached to an IV (intravenous) line placed into one of the patient's veins. A computerized pump attached to the IV lets the patient release pain medicine by pressing a handheld button (Figure 13). PCA therefore enables self-administration of predetermined doses of analgesics to ease the pain and thus enhance the patient's comfort during the treatment.

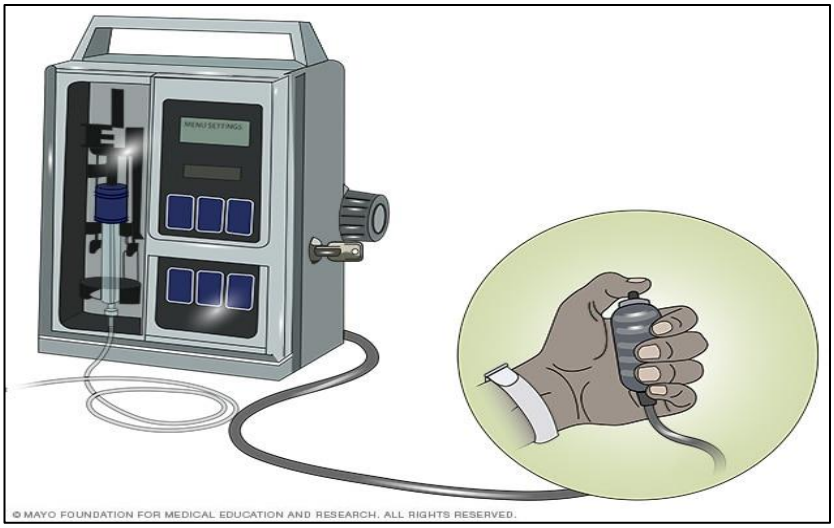


Figure 13 Infusion pump with handheld device so that the patient can require bolus (extra dosage).

PCA IV treatment demands several procedures and not all patients are eligible for the treatment. Initially, doctors define the amount of opioids that the patient needs for the treatment, usually combining drugs of various intensity and criticality with the

respective dosage. Furthermore, they might evaluate whether the patient is eligible for the treatment. To do so, doctors need to examine the patient's behavior during surgery, age, weight, apnea history, and so on. If they deem the patient eligible for this treatment, they define the dosage for the infusion pump programming. The patient and their family need to be trained by the clinical team in order to learn how to identify the pain score and understand the safety instructions and how the pump works. The ward team also need to be prepared to periodically monitor the patients in treatment so that they can properly identify sedation levels, pain score, vital signs, respiration status, etc. Hence, the ward team, the doctors, and the patient need to cooperate in order to realize safe and effective treatment.

Despite the various benefits, and although PCA is becoming one of the most effective techniques for treating postoperative analgesia, several institutions have reported accidents associated with PCA treatment. The American ECRI [1, 2, 51, 52] institute monitors health technology hazards in order to inform healthcare facilities about safety issues involving medical devices and systems. They provide an annual report listing the top 10 hazards as the potential sources of danger that they believe to warrant the greatest attention for the coming year . In the last seven years, overdosage caused by IV infusion pumps has appeared in the list five times.

One fundamental cause of problems is that opioids (such as sufentanil, piritramid, morphine, remifentanil, fentanyl, and others) have respiration depression as their main side effect. Other reported causes are related to human errors in drug prescription, wrong programming of the infusion pump, and PCA-by-proxy (another person requests a dosage for the patient).

In light of these facts, several organizations have made decisions to address the increasing occurrence of accidents. For example, the FDA launched the Infusion Pump Improvement Initiative to prevent infusion pump problems by (a) establishing additional requirements for infusion pump manufacturers; (b) proactively facilitating device improvements; and (c) increasing user awareness. Moreover, several works and health associations [142, 147, 156] have urged healthcare professionals to consider the

potential safety value of proper monitoring, for example of oxygenation and ventilation, in patients receiving IV opioids during the postoperative period. [142, 146, 147, 156], have shown the relevance of safer patient monitoring in PCA treatment through the combination of different types of devices such as pulse oximeters and respiration sensors (capnometry). Meisenberg et al. [161] claim that effective monitoring needs a wider evaluation of vital signs and patient history. Hence, the literature and standards have provided solutions for enhancing system awareness and then enhancing monitoring capabilities to mitigate the hazard of overdosage.

4.1.2 Medical cyber-physical systems for overdosage mitigation

The specification of MCPS architecture has evolved in recent years along with demands and challenges. [77, 93] defined requirements and an abstract architecture for defining a medical cyber-physical ecosystem which encompasses applications, cooperative medical devices, and integrated hospital information systems. Their work provided the basis for the standard ASTM F2761 [72]. This standard specifies general requirements, a model, and a framework for integrating medical devices to create an Integrated Clinical Environment (ICE). The ICE shall behave as a single system via safe integration of all the equipment from different manufacturers in order to improve patient safety, enhance treatment efficacy, and increase workflow efficiency. Moreover, the standard defines a catalog of use cases (clinical contexts and clinical scenarios) for the application of medical cyber-physical systems.

In this work, we consider the safety interlock scenario from the ASTM F2761 standard for different applications, such as intensive care units, medical surgery, pediatrics, neonatal care, etc. The main aim of the safety interlock scenario is to avoid respiratory depression caused by excessive opioid doses in patient-controlled analgesia treatment. This can be realized by the continuous monitoring of physiological parameters such as oxygen saturation (SpO2, which can be measured via pulse oximetry), heart rate, respiration rate, and end-tidal carbon

dioxide (EtCO2, which can be measured via capnometry/capnography). Hence, once the ICE identifies any respiratory distress signal, it autonomously sends a stop command to the infusion pump and sounds an alarm for the caregivers. System configurations vary according to the patient status assessment strategy and the respective monitoring systems, which may include:

1. **Configuration 1** - *pulse oximeter* – In [13, 173], the authors consider only the pulse oximeter sensor for monitoring SpO2 (blood saturation oxygen) and heart rate (Figure 14 Area 1);

2. **Configuration 2** - *respiration sensors* - [147], the authors used capnography sensors to collect respiration rate and end-tidal CO2 data (EtCO2) (see Figure 14 Area 2);

3. **Configuration 3** - *both combined* – In the standard ASTM F2761, it is proposed that both data from respiration sensors and from a pulse oximeter should be combined to detect signals of respiratory failure (see Figure 14 Area 3).

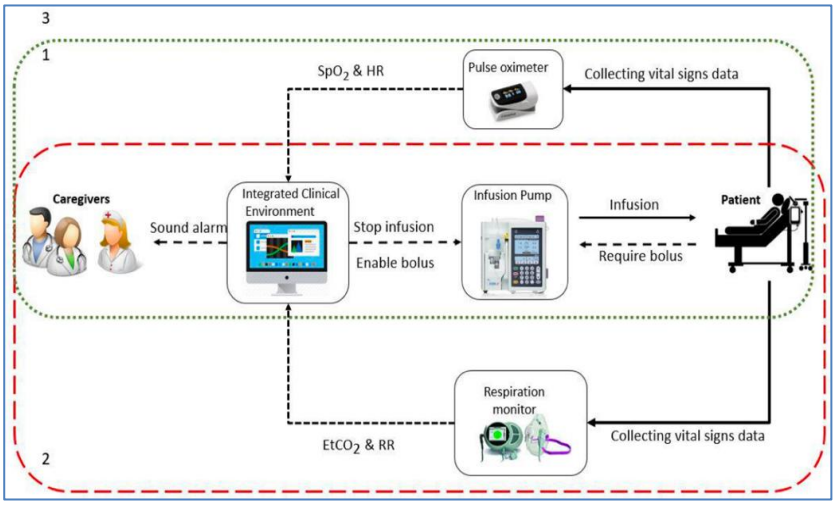


Figure 14 Potential system configurations for joint functions of smart PCA infusion

Figure 15 shows a UML sequence diagram describing the expected system behavior for the safety interlock scenario. It is assumed that this is the most complete configuration (Config. 3 - both sensors), with the sensors monitoring the patient's vital signs

through physical connection to the patient's body, such as pads and cannulas. In the figure, all method calls (monitorHeartRate, monitorRespirationData, monitorSpO2, and monitorEtCO2) to the patient consider that the physical connection to the patient is set and provides the required information to the devices. All devices send data to the ICE manager through calls to the update methods, which periodically update the vital signs data from the patient to the ICE. It is worth noting that the efficiency of the ICE manager depends on the device's update frequency which varies by manufacturer and by the quality standards that the devices implement. The ICE Manager evaluates whether the vital signs violate any thresholds previously defined for the patient's treatment and then triggers the stop infusion in case any violation is detected. Finally, it calls the method notifyClinicalStaff to sound an alarm for the caregivers through the alarm systems.

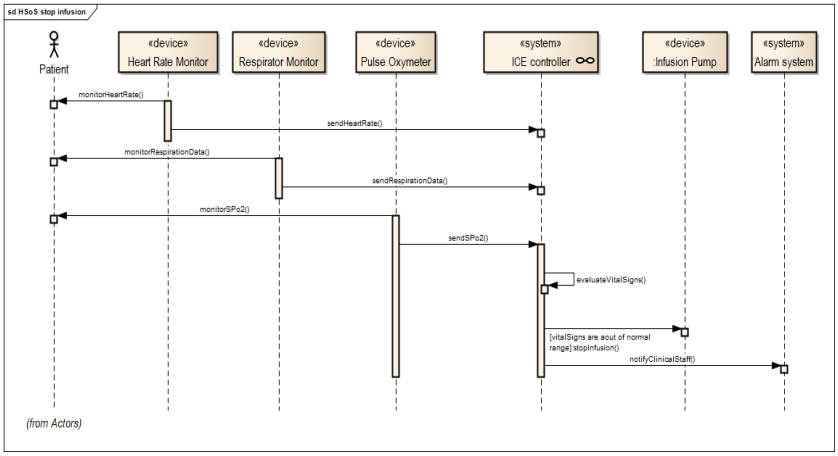


Figure 15 Sequence Diagram for the feedback closed-loop infusion system

Figure 16 depicts the enable bolus scenario defined by Pajic and Arney [173]. It has the same monitoring procedure, but the actuation is positive instead of being prohibitive as in the case of the safety interlock scenario. The sensor devices monitor the patient's vital signs and send the updated data to the ICE manager. In this way, when no violation in the predefined thresholds is detected, the ICE Manager calls the enable bolus function in the infusion pump so that it can update the token and let the patient have the bolus infusion. Finally, when the patient requires the bolus infusion, the pump will only give the opioid to the patient

if the token is updated by the ICE Manager's enable bolus function.

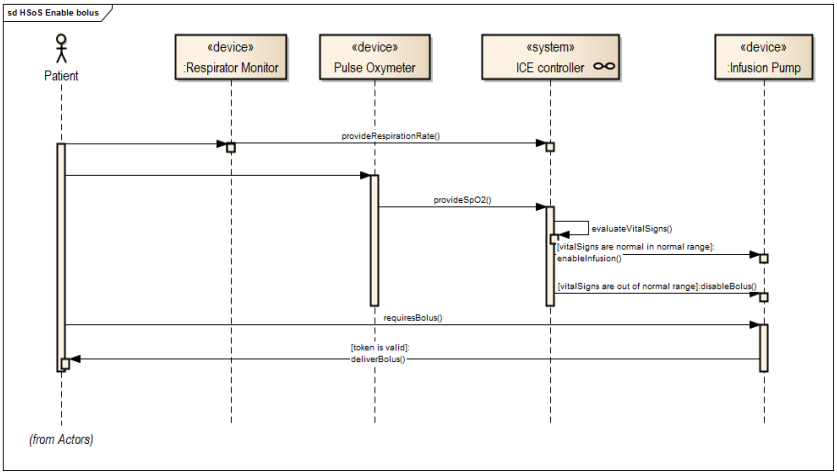


Figure 16 Sequence diagram for the enable bolus scenario

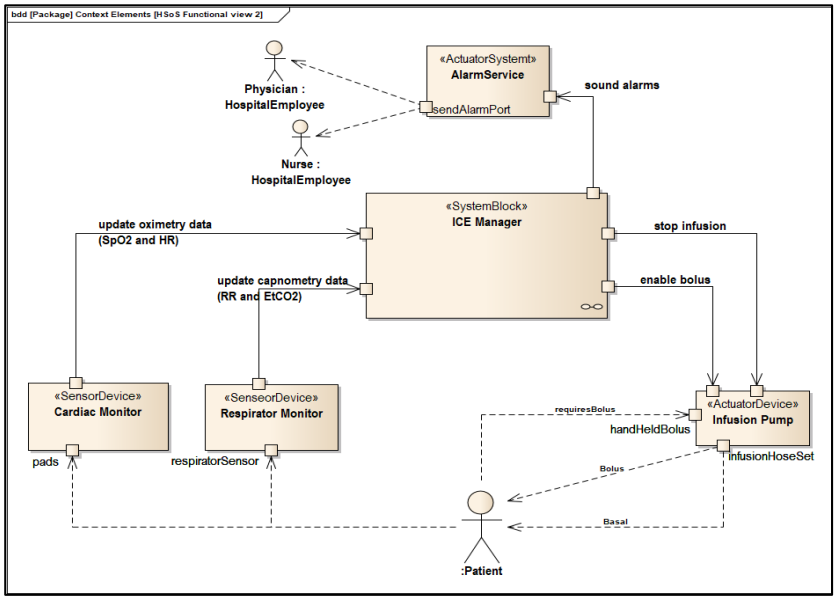


Figure 17 Component network for the use case scenario

The component network in Figure 17 represents the static interaction between all potential components of the system and the context for the most complete configuration setup. There might be system adaptations using only a pulse oximeter, or only a respiration sensor, or a variation of multipurpose devices such

as multiparameter sensors. Hence, we focus on the main services provided by the devices.

In Figure 17, the sensor devices collect the patient's physiological signals through physical interfaces, which also need to be connected to the ICE Manager through network interfaces. It is worth noting that for the enable bolus and/or stop infusion functions we might have variations of the currently connected sensors. Figure 17 shows the most complete configuration. We can also observe the two control functions (enable bolus and infusion stop) provided by the infusion pump so that the ICE Manager can control the infusion operation according to the current situation. It is also assumed that the alarm system is independent of the ICE Manager and needs to provide interfaces for sounding the alarms according to the situational risk level.

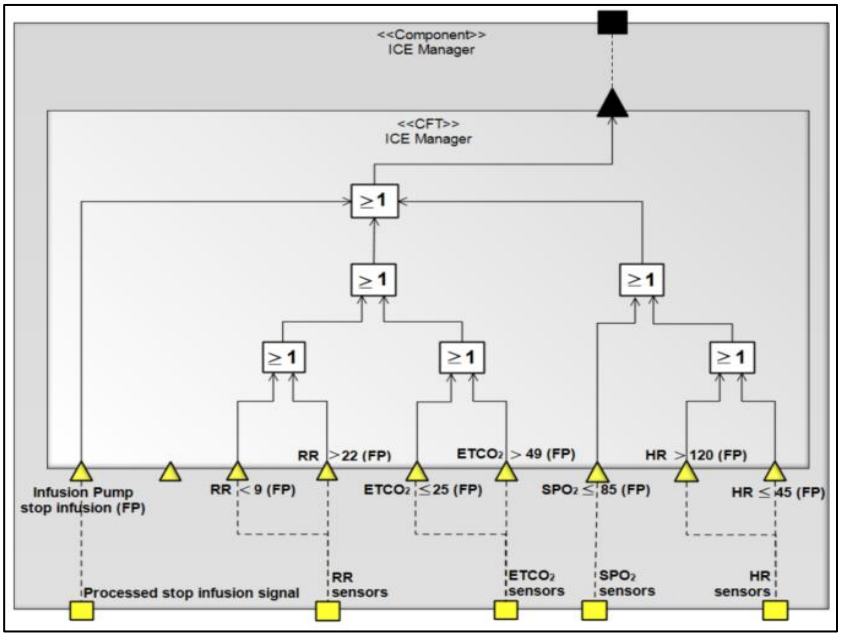


Figure 18 Stop infusion command omission CFT, describing failures of the ICE and medical devices orchestration.

The hazard being analyzed is: **opioid overdosage when the patient's heart rate is lower than 50 beats/min or greater than 120 beats/min OR SpO2 is lower than 90 percent OR respiration rate is lower than 10 breaths per minute OR**

EtCO2 is greater than 60 mmHg. We can see that such a hazard might be caused by both of the described functions: stop infusion and enable bolus. The former can enable the analyzed hazard through the omission of the *stop infusion* command. The latter can enable the given hazard through the commission of the *enable bolus* command, giving the opioid when the patient should not have it. However, both failures can occur due to the activation of similar failure modes analyzed in the CFT presented in Figure 18.

Eight intermediate faults, when combined in particular ways, can cause the stop infusion omission (or the enable bolus commission). We simplified the complexity of every single intermediate fault resulting from the distributed characteristic of the overall system. The sensors and actuators are remotely connected to the system through a network. The medical devices measure the patient's vital signs and transmit the data to the ICE. The ICE detects the new data that arrives, assures the correctness of the gathered data, and evaluates the risk according to the data. Thus, for every single sensor's fault, we assume that something wrong occurred between the vital signal measurement in the device and the final data verification in the ICE. For instance, there are two possible intermediate faults that might occur for the respiration rate sensor ICE component: when the respiration rate gets lower than 9 BPM or higher than 22 BPM. Overall, there is a difference between the real vital signs in the patient's body and the system data.

For the infusion pump omission intermediate fault, we assume the same connectivity of the sensors and an omission that might be caused by wrong implementation of the ICE driver responsible for sending the command to the infusion pump, by a failure in the network, or by an error in the infusion pump. We also assume that the infusion pump works with both remote commands: *stop infusion* and *enable bolus*. In the infusion pump, the command *stop infusion* needs to be performed immediately as soon as it is received by the device. To enable the bolus, the device needs to check if it has received an available token previously sent to the pump before giving the opioid to the patient.

Mitigating these failures implies performing an analysis of all the medical devices constituting the system. However, these medical

devices will only be known at runtime when the healthcare provider (hospital or clinic) integrates the available sensors and actuators. Moreover, hospitals usually work with different types and brands of devices. This variety might imply different guarantees of service quality provided by the medical equipment. Furthermore, in the healthcare provider environment such systems are susceptible to various changes in their components due to maintenance issues, patient ergonomics, and so on.

4.2 Drivers for completeness of risk assessment

The underlying concept of risk assessment is captured by the National Academy of Science's "Red Book" [180], where assessment of risk is treated as a scientific activity limited by the available knowledge and the uncertainty inherent in risk. The main aim of risk assessment is therefore to structure, through systematic modeling, the information and knowledge available at the detailed component/basic event level in order to assess the accident risk at the system level.

For the purpose of formal analysis, there are several quantification formulas for representing and characterizing the concept of risk so that an objective analysis and evaluation can be done and assured. According to ISO/IEC 14971 and ISO/IEC 61508, the main risk components are twofold: the probability of occurrence of harm; and the consequences of such occurrence of harm. However, both the probability and the consequences of harm strongly depend on the chain of events leading to an accident. Therefore, an effective risk assessment technique needs to (a) identify all the usage scenarios and (b) analyze their structure (chain of events) to check in which situations some hazardous event might lead to harm.

For risk assessment in practice, this definition leads to the need to develop methods for the identification of the complete set of accident scenarios that might occur, and for the accurate estimation of their frequencies of occurrence and their consequences. As completeness of scenarios cannot be guaranteed and accuracy of estimation must be evaluated in the face of limited knowledge and approximated modeling, a

definition of risk needs to be introduced to account for the uncertainties associated with risk assessment:

$$Risk = \{s_i, p_i(f_i,\ c_i)\}, i = 1, \dots N + 1 \quad (1)$$

Where s_i represents the sequence of events of the i-th of N accident scenarios, f_i represents the frequency of occurrence of such a sequence of events, c_i is the consequence that would result if that scenario were to occur, and p_i(f, c) is a joint probability density function describing the uncertainty in the frequency of occurrence f_i and the consequences c_i of accident scenario s_i. The N+1 scenario is added to account for the incompleteness of the set of scenarios, i.e., for those scenarios not considered because they were unknown at the time of the analysis (i.e., the so-called "residual risk").

Hence, according to equation (1) and the main standards [90, 95, 255], assuring completeness for the Hazard and Risk Analysis depends on three elements: (a) functional completeness; (b) operational scenario completeness; and (c) hazard completeness.

Functional completeness can be defined as the "*degree to which the set of functions covers all the specified tasks and user objectives*" [96]. A fundamental assumption for traditional HaRA techniques is a well-defined and complete specification of the system under analysis. Moreover, considering the system behavior as what the system does to implement its functions, classic safety engineering methods also assume as essential input a concrete and well-defined architecture specification [14]. Hence, both function completeness and architectural specification underpin the whole process of the Hazard and Risk Analysis and directly affect the completeness of the analysis.

Operational scenario completeness depends on the identification of all usage scenarios for the system. To derive this information, engineers need to specify the system context, the users, other systems, interactions, and foreseeable situations. For medical cyber-physical systems, achieving completeness of operational scenarios might be an endless task due to the variety of actors, interactions, system changes, and innumerous possibilities for the systems to fulfill the needs of patients and caregivers.

The hazard and risk assessment process needs to identify all the relevant hazards that might arise during usage of the system in the operational scenarios. Safety engineering defines several techniques focused on system function failures, or misuse deviations [54]. Failure identification coupled with structured scenario descriptions andthe use of guidewords provides a systematic method for failure identification that assures a certain level of completeness. Identification of potential failure causes provides useful information to designers when determining optimal ways of preventing or mitigating the effects of failures [54, 83, 141, 157, 199, 243].

4.3 Enhancing completeness for risk parameter elicitation

Considering the related drivers and challenges for the completeness of risk assessment, we derived an approach for enhancing the completeness of risk paramenter elicitation for medical and cooperative CPS. A fundamental element of the proposed approach is a meta-model that structures the required multidisciplinary knowledge for deriving risk parameters.

The backbone of the model is the risk definition previously defined in equation (1). Hence, we started from the main risk concept and then refined and classified the risk parameters according to the main aspects of risk assessment and its main elements, namely system elements, context issues, and functional safety. Context issues represent relevant effects that the system actuation might cause in the environment, for instance regarding involved caregivers, patients, procedures, and actuation environment. System elements organize and classify the concept of system behavior, system function specifications, the architecture, provided services, potential adaptations, and components. Functional safety models all the concerns regarding potential system failure guarantees exhibited by the system components and their abstract configurations.

In the next sections, we will describe this meta-model in detail. The meta-model is independent of the implementation technology used; however, we instantiate the approach for developing a Bayesian Network (BN) model for evaluation aspects. Other approaches could also use our technique, as any other calculation

approach also needs to know which parameters affect risk. In a BN-based calculation approach, each parameter becomes a node in the BN. The challenge is then to build an overall BN based on these initial nodes. The meta-model provides crucial input for coping with this challenge as it shows the relationship between risk and risk parameters.

4.4 A metamodel for risk parameter elicitation

The elicitation of the risk parameters has to handle multidisciplinary and (sometimes) conflicting concerns from different stakeholders. Based on our experience in dealing with people from diverse educational backgrounds, we derived a meta-model for the identification and classification of a wider range of relevant risk parameters to enhance completeness w.r.t concerns such as risk assessment, domain-specific data, system-specific data, and reliability data. In this section, we will explain the elicitation process using the risk meta-model applied for the case study presented in the chapter 4.1.

Figure 19 depicts the meta-model for eliciting the risk parameters. The model expresses the relationships (using UML class diagram language) between the concepts, which are concisely organized into four abstraction layers. At the top of the figure, we placed the main abstraction layer called “Risk Assessment”, which contains all the other layers in the lower part of the figure: “System in its Usage Context”, “System Functional Realization”, and “Functional Safety Analysis”. Every single layer will be detailed in the next subsections.

4.4.1 Risk Assessment layer and System in its Usage Context layer

The Risk Assessment layer is the main meta-model layer that supports the safety engineer in thinking about everything that affects risk estimation and evaluation. Considering the risk definition presented in the equation (1) , this layer helps to define the joint probability density function based on the frequency of occurrence and the consequences. These function elements are derived from the scenario analysis and can be classified into system elements and context elements. System elements drive the identification of a system’s hazardous behavior through the states

that the system might assume; the system elements then define all the relevant properties needed to describe a system state. Likewise, context elements support the analysis of how hazardous behavior can cause harm, so the context can be characterized by the set of relevant properties for identifying the current environment state.

The potential traces to harm are the basis for measuring risk at runtime. In a concrete runtime situation, we must analyze which traces might occur and their likelihood. We assume that severity was included in the previous hazard analysis where the considered accident was identified; thus we consider probabilistic elements related to the potential harm with its inherent severity. Both severity and probability might oscillate according to the current runtime situation; however, a worst-case assumption has been reasonable in all the scenarios we considered so far. We can assume this due to two factors: (i) The overestimation of risk is strongly limited if we work with worst-case assumptions; (ii) additional complexity might cause underestimation of risk.

At the top of Figure 19, we present a UML model that represents the risk concept mentioned above. The Risk Metric entity realizes the idea of a risk metric capable of assessing the risk of the situation by aggregating the system-related elements (left side) and the context-related elements. We decided to make the context elements optional in order to keep the model compatible with legacy risk metrics that do not consider context elements.

This risk assessment layer is a conceptual representation of a risk function that needs to be implemented and deployed in a monitor for runtime risk assessment. This conceptual model assumes that any technology can be used for realizing the risk metric. The literature and our experience show that tree models tend to capture the hierarchical characteristics of the risk model better.

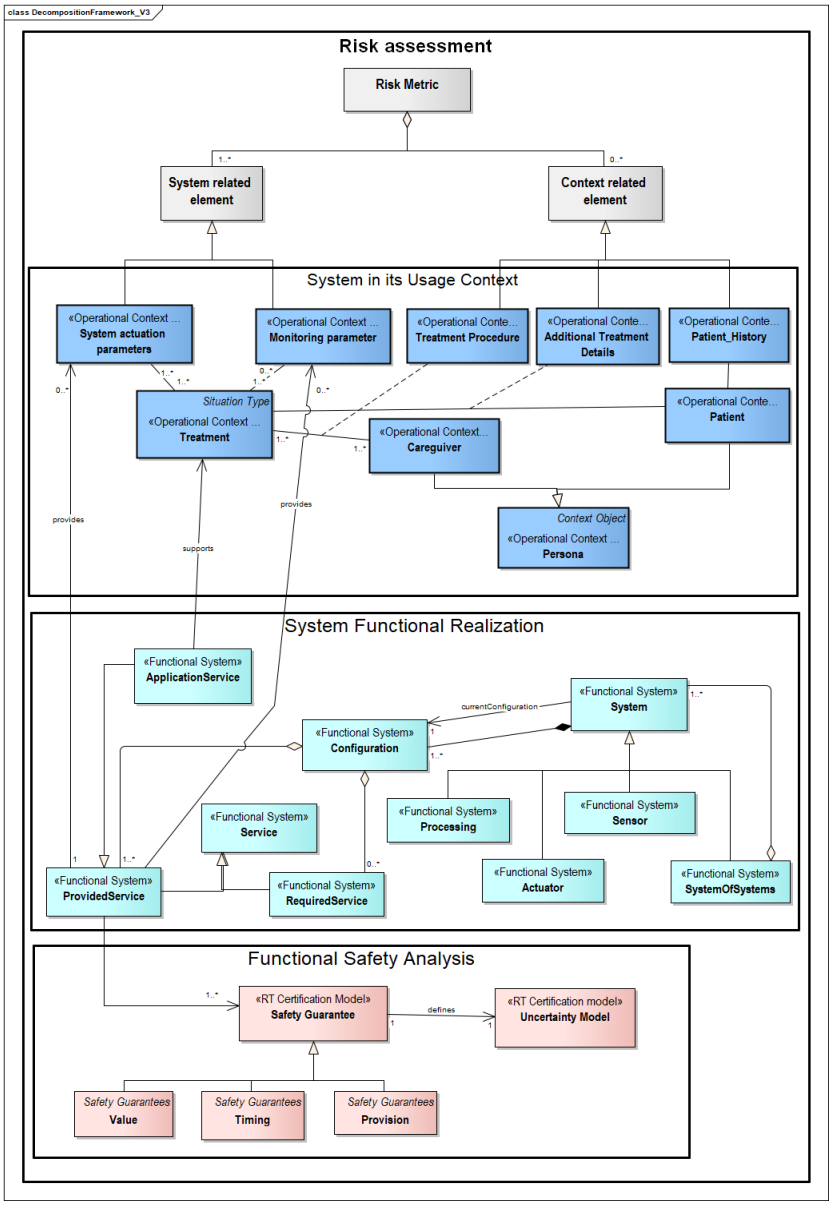


Figure 19 Risk meta-model

According to Zio [248, 251], the Bayes theorem provides a common framework for describing the uncertainty in the risk assessment. Moreover, the framework is suitable for encompassing all expert opinions can be combined with statistical data to provide quantitative measures of risk. Furthermore, Bayesian networks have been adopted as the main model for

implementing reasoning about uncertainty and probabilistic risk detection by a wide range of domains, such as nuclear power plants, chemical plants, autonomous marine systems, and connected health [3, 25, 105, 228, 248].

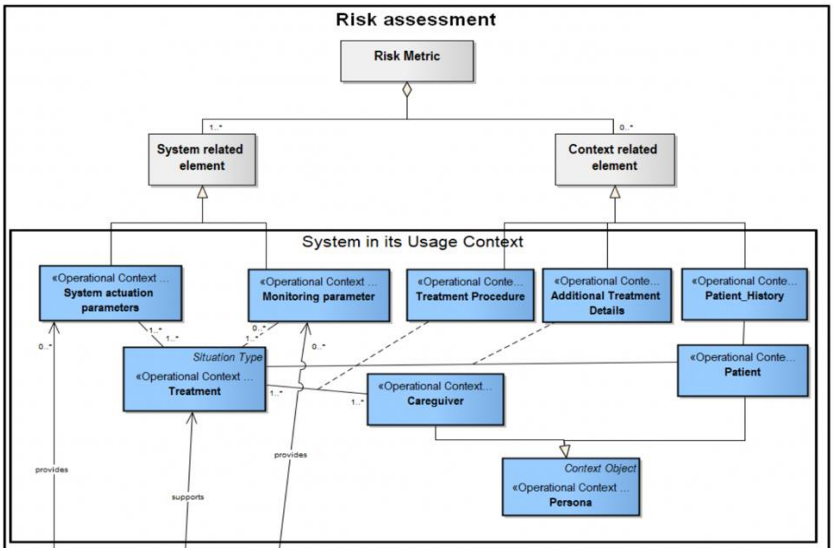


Figure 20: Upper layers of the risk assessment model. The top presents the main classification for the risk assessment parameters; the second level shows the System in its Usage Context layer, which refines the risk parameters.

The second abstraction layer ("System in its Usage Context" in Figure 20) refines the system and context entities from the top layer to identify a set of elements that represent relevant domain-specific data. This layer is responsible for narrowing down the scope of the application domain and is thus totally dependent on the body of knowledge gathered by the safety engineers. In this sense, safety engineers need to define this layer according to standards, experts' requirements, and the specific literature. Hence, we present this layer based on the body of knowledge gathered from the medical domain. To establish this layer, we performed interviews with medical experts, investigated the literature, and studied hospital procedures and various medical systems safety standards, such as EN ISO 14971:2012, IEC 62304, ISO 13485, ASTM F2761-2010, and the IEC 60601 series. Furthermore, we cataloged a set of potential application scenarios for medical cyber-physical systems to enable us to define a relevant set of potential accidents and hazardous situations for medical CPSoS.

We derived this layer using a system abstraction level of reasoning where the system is not detailed in its building blocks but where the focus is on the overall behavior. Therefore, considering the literature and the medical experts opinion, we classified all the operational context elements according to the following structure:

1. a set of patient-monitoring parameters;
2. elements related to the system actuation, such as infusion, sounding alarm, stopping any treatment provided by a medical device, etc.;
3. identification of all caregivers responsible for the treatment and their related procedures;
4. additional treatment details concerning the patient's healing, such as any additional medication or procedure that might affect the treatment; and
5. patient's history data such as weight, age, any relevant history of disease.

It is worth noting that this abstraction layer is interchangeable and reusable. Stereotypes guide the classification of relevant data elements in this layer. As can be seen in Figure 20, there are Operational Context stereotypes that encompass details about the Situation (according to the Situation Type) and Context Objects that participate in the scenario. Hence, this abstraction layer provides relevant inputs not only for this approach but also for any other elicitation process that needs to identify such relevant domain-specific information.

4.4.2 System Functional Realization layer

In [132], we proposed an extension of Laprie's taxonomy for defining new concepts for cooperative systems of systems that were not specifically defined in the taxonomy.

Cooperative systems are complex system composed of multiple and dynamic entities that cooperate (through information sharing or tasks) to achieve a common objective [74, 80, 169, 217]. An application service is responsible for realizing the final common objective. However, this service is provided by the whole

cooperation between different services established and thus only known at runtime. We define the term "**joint service**" to specifically target application services provided by a cooperation of services in a system of systems. Every single entity in a cooperative CPSoS can provide two types of services: **social services** and **individual services**. Social services are those that are provided in a service orchestration to realize an application service and/or joint services. The "individual service" delivered by a system (in its role as a provider in a cooperation) is its individual behavior as perceived by its user(s), since all the entities in a system of systems are independent and also have their own proper functions and objectives.

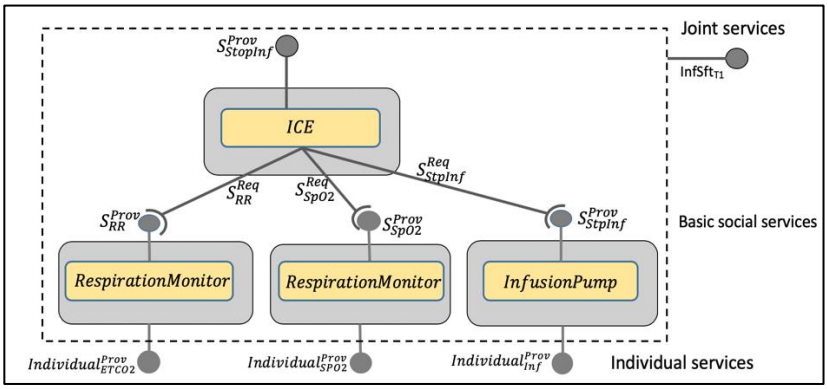


Figure 21: Concept of how the different types of services are provided in a cooperative system-of-systems environment.

Figure 21 illustrates how the conceptual services definitions cooperate in the environment to accomplish the objectives. We can observe services delivered by independent devices in a medical CPSoS, namely RespirationMonitor (provides social and individual services for respiration rate); PulseOximeter (provides social and individual blood saturation data services – SpO2); InfusionPump (provides social and individual infusion service); and the Integrated Clinical Environment (provides only the social application service "stop infusion" based on the cited required services). At the top of the figure, the joint service (infusion safety) is depicted, which is provided by the current system configuration through the cooperation between all provided and required services mentioned above.

We defined the elements of the System Functional Realization layer based on these concepts defined for cooperative systems. The layer provides the structure for identifying how the elements of the System in its Usage Context layer are provided by the systems/devices and their potential configurations.

In Figure 22, the System entity represents an element of the SoS. It can be a single entity such as a processing node (ICE), an actuator device (infusion pump, defibrillator, etc.), a sensor device (respiration monitor, pulse oximeter, multiparameter monitor, etc.), or a complex structure composed of other system entities in the case of a system of systems. Every system might have different configurations defining the current set of exposed services. It is worth noting that single entities of the system have concrete configurations, whereas complex SoS provide abstract configurations, which are virtual configurations specified at design time that are only realized completely at runtime. Services are characterized by their current set of provided guarantees, which are defined by several attributes related to safety, reliability, performance, security, and so on. Therefore, the same system often has diverse configurations providing services with different guarantees, especially for configurations of social and individual services.

The left bottom of Figure 22 shows the ProvidedServices entity, which realizes social and individual services for different systems and configurations. Provided services expose their safety guarantees attributes (to be defined in the next subsection) according to their current behavior. Monitoring and system actuation parameters (from the upper layer Systems in its Usage Context) can be sensored and provided by services, respectively be implemented by sensors or actuator devices. Furthermore, an application service is a special type of provided service which supports a treatment and is implemented by a cooperation of services.

The RequiredServices entity is responsible for implementing a service demand so that another service can fulfill its requirements and then a provided service can be performed by a cooperation.

We define a system configuration as a set of both provided and required services that cooperates to accomplish a specific high-level function (application service) such as safety interlock. We also assume that all context-related information regarding treatment details and patient data is provided by information systems supporting this infrastructure.

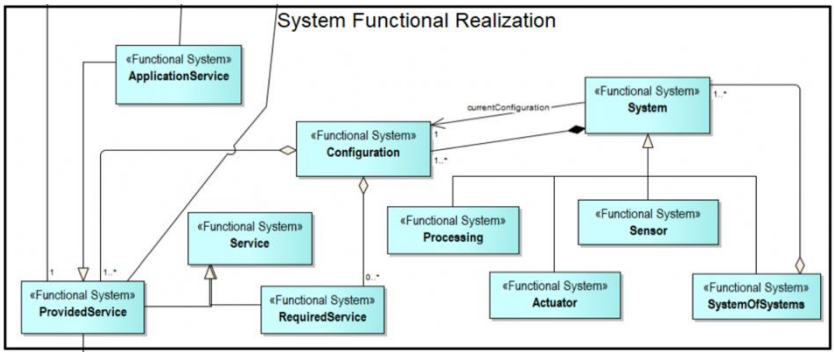


Figure 22: System Functional Realization layer

4.4.3 Functional Safety Analysis layer

In the Functional Safety Analysis layer, we specified all the safety guarantees (accuracy, provision, and timing guarantees [8, 48, 87, 175]) exposed by the system services. We assume that such guarantees are specified through formal languages such as ConSerts [203], which shows reliability data defined at runtime through machine-readable certificates. Each service hence exhibits an uncertainty model for its behavior; for example, pulse oximeters show different degradation levels while measuring blood saturation, as their accuracy varies according to the measurement technique used and/or the system implementation details, as you can see in the Figure 23.

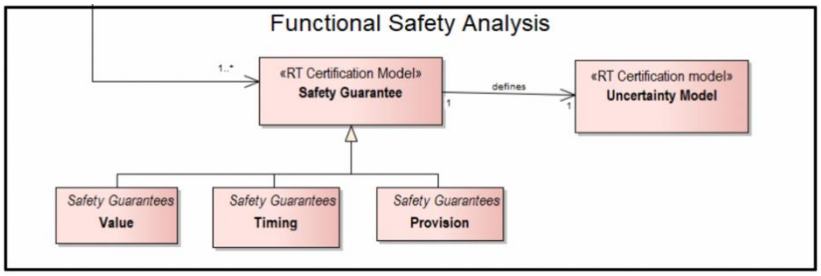


Figure 23: Functional Safety Analysis layer

4.5 Risk parameter elicitation according to the meta-model

The main aim of the meta-model is to underpin a structured approach for enhancing the elicitation of relevant risk parameters. The approach is derived based on the meta-model to guide safety engineers in managing all the multidisciplinary knowledge in order to derive risk metrics for runtime risk assessment. In this subsection, we present the tasks and concepts of the approach and, for the sake of understanding, apply the concepts in a well-defined problem related to medical cyber-physical systems of systems in subsection 4.1.

Implementing dynamic risk assessment for CPSoS implies dealing with context and system changes. This approach initially starts by deriving the most complete set of relevant risk parameters in order to enhance context awareness and then focuses on the impact of system adaptation changes on the risk metric. Hence, the final result of the approach is a so-called runtime risk model, which encompasses not only a risk metric but also defines how the risk metric needs to adapt according to the system changes.

Furthermore, realizing dynamic risk assessment calls for implementing a concrete model capable of assessing the risk of the current situation. This approach therefore provides support for building a Bayesian Network due to the reasons mentioned above. However, it can be easily adapted for integrating any other technology, since it conforms to any hierarchical structure that can perform probabilistic risk assessment based on a set of parameters.

The approach follows the top-down structure defined by the meta-model in order to derive risk metrics for situation assessment. In Figure 34, we present a complete view of the approach for deriving the dynamic risk model. It is worth noting in the diagram the five macro activities that define the activities related to risk parameter elicitation, risk model building, and risk model adaptation, as described in the following subsections.

4.5.1 Initialization of the risk metric structure

Once the meta-model has defined a hierarchical structure for parameter elicitation and risk model building, the first activity is focused on the instantiation of the root nodes of this tree structure in order to derive the whole risk metric.

In Figure 24, we highlight all the tasks related to this activity. The first task refers to the creation of the main root node, which refers to the main node defined in the meta-model's Risk Assessment layer. It is worth noting that these nodes also need to define a quantification strategy based on the implementation technology for quantitative risk assessment.

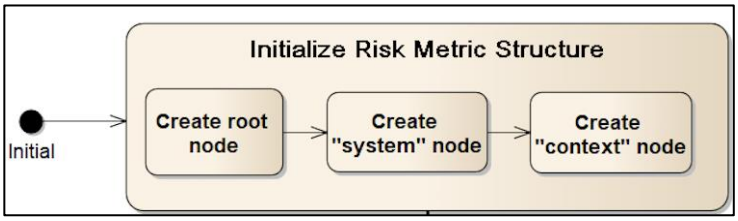


Figure 24 Initialization of the risk metric structure

Once all the required information has been arranged, the creation of the root node must represent the risk of an overdosage to the patient. As stated in Section 4.1, defining the overdosage event is complex, as it is necessary to clearly define boundaries to determine when such an event actually occurs. We thus model the event as an uncertain variable that expresses the probability of an overdosage occurrence. Due to the complexity of defining the severity of the harm, we decided to be conservative in identifying when the event is about to happen.

The quantitative aspect for the root node can be defined by applying Bayes theorem to integrate all the observations from the different sources. This yields the following equation for defining the CPD table:

$$P(z_1, \dots, z_n \mid x) = P(z_1|x) \dots P(z_n|x) = \prod_{i=1}^{n} P(z_i|x)$$

Equation 3 Probability equation for ach CPD

Where all the variables z_i represent observations from the independent sources with individual likelihoods. We also assume that a given state $x \in X$, where X defines a set of potential states in which the situation might be classified. Therefore, equation (2) defines the probability of an eminent overdosage state x that we want to detect. The overall risk nodes need to be refined depending of the implementation technology utilized for modeling the risk. For example, in Bayesian Networks, the process for defining the current probability of a hazardous situation being activated can be adjusted according to the experts' knowledge modeling and/or historical data.

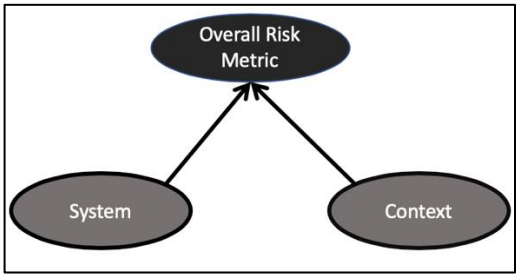


Figure 25 First derivation of the risk metric

In Figure 25, we show the first representation of the model built after the first derivations of the system and context nodes. The two child nodes calculate the aggregated risk at different actualization rates. The System node must deal with online system device data from sensors and actuating devices. The Context is required for dealing with data gathered from context monitoring. The update speed is not high due to the sensoring aspects, which depend on manual interference and/or data that does not change within a short of period of time, such as medications and clinical evaluations.

A Bayesian Network representation can thus be initialized to represent the concepts explained above. In Figure 26, we have a representation of all the aforementioned nodes and the possible elements for constructing the node probability tables (NPT). NPTs can be defined by a classical five-degree representation of risk, such as negligible, minor, serious, critical, and catastrophic. However, critical and catastrophic states are often understood as situations where the unwanted event has just occurred and the difference between them is the severity of the harm. On the other side, negligible and minor situations are deemed acceptable risk situations and often no or only minor risk control actions are required to be performed for these situations. Finally, serious situations are considered as warning or alert situations. They represent the eminent triggering of the unwanted event and risk control actions are frequently required for these situations.

We decided to keep a three-state classification for BN NPT due to practical reasons. First, we assumed a conservative risk assessment for the evaluation of situations. Moreover, we want to predict trends for hazardous situations so that the unwanted event can be avoided by the risk control actions. Therefore, considering the meaning of the critical and catastrophic risk levels defined above, they are not useful for deploying risk control measures. Once the harm has occurred, the system is unable to bring the patient back, given the specified role of the actuation devices in the overall system configuration.

Moreover, factorial explosion of NPTs is an important issue to be avoided during the building process of BN. Hence, whenever possible, we decided to keep the three-state situation classification.

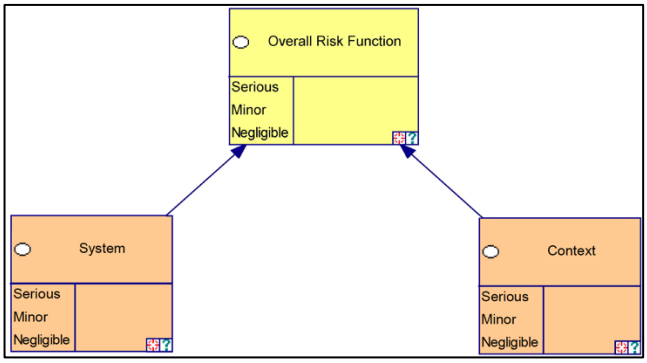


Figure 26 Bayesian Network for the risk metric

In the following subsections, we will define which observations need to be modeled and structured for the Bayesian Network. We will first show the refinement of the System node and then the refinement of the Context node in the next subsection.

4.5.2 Refinement of the System node

The System node models the risk measured by all the entities that are related to the system's state (see Section 4.4), which means that a quantitative assessment model must calculate the aggregated risk for the current state from the data gathered by the sensors and actuators and the active system configuration.

First of all, as input for this process, the safety engineers need to organize all the information about the treatments, the monitored parameters, and the actuation mechanisms, as described for the System in its Usage Context layer. For example, for monitoring the risk in patient-controlled analgesia treatment, the exhaled carbon dioxide is a more reliable indicator than monitoring the level of blood oxygen saturation. Moreover, engineers must define all the possible devices (processing, monitoring, and actuating devices) that might take part in the system as soon as they got to know the potential abstract configurations and possible transitions. Hence, they can seize the effects of these changes during the definition of the risk assessment model.

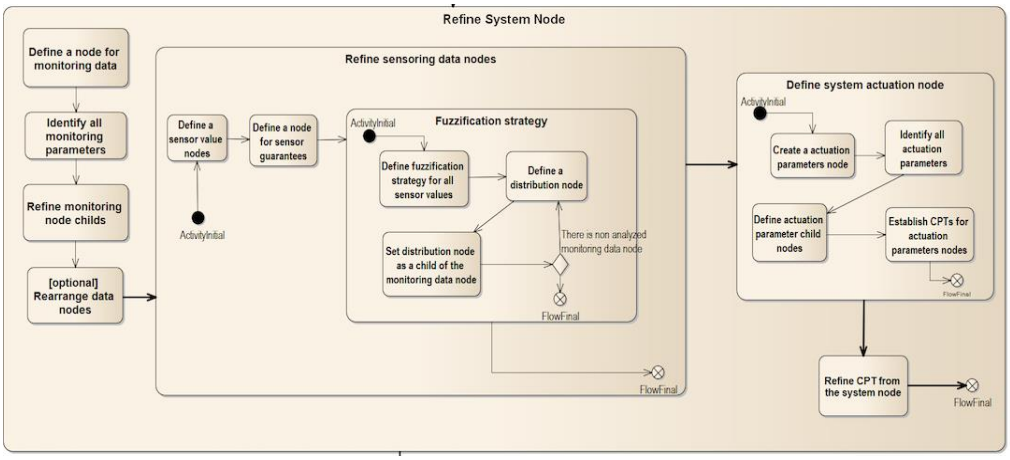


Figure 27: Activities related to the refinement of the System node.

The first four activities in Figure 27 concern the definition of a node to represent the aggregated assessment based on the system-monitored data. In this sense, the safety engineer identifies the monitoring parameters as specified in the Systems in its Usage Context layer meta-model and creates a tree structure to represent the monitoring. For example, for the situation assessment for avoiding an overdosage event, blood saturation, heart rate, exhaled carbon dioxide, and respiration rate need to be monitored. Hence, we instantiated a representation for all monitoring data as shown in Figure 28.

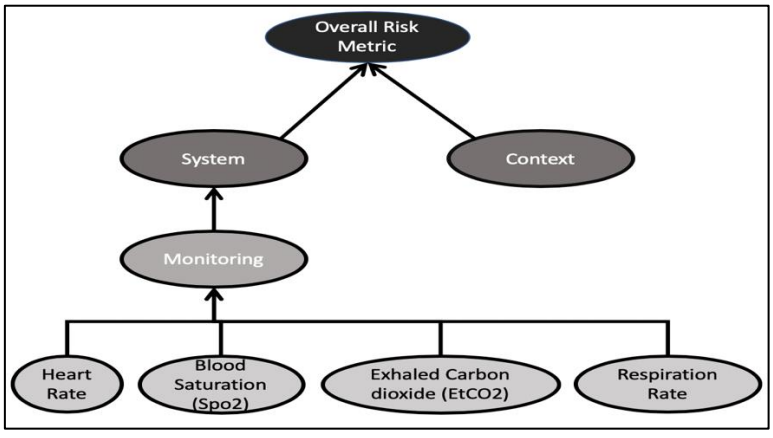


Figure 28 The refined BN structure for the System node with the Monitoring node and its data nodes modeling the parameters

After identifying a preliminary node structure, we can refine the structure in order to simplify the NPT of the Monitoring node and

avoid factorial explosion of its NPT. We can apply, for instance, the divorcing strategy to establish a second node layer to aggregate data for pulse oximetry (heart rate and blood saturation – SpO2) and capnometry (respiration rate and exhaled carbon dioxide – EtCO2). Figure 28 depicts these data nodes, which combine the monitoring data nodes and keep the NPT of the Monitoring vital signs node simple enough. We decided to implement the NPT of the aggregation nodes as an OR-function in the BN and also preserved the five-class discretization defined by medical experts for the monitoring values.

Once we have defined the node structure and the node probability tables (NPT) for the nodes at the top, it is necessary to establish the probability tables for the nodes at the bottom of the tree. For the monitoring parameters (leaf nodes at the bottom of the tree in Figure 28), we decided to keep the five-class classification of risk specified by medical experts based on the discretization of the continuous values that every single parameter might assume. For example, to monitor blood saturation (SpO2), the scale is a percentage of saturation from 0 to 100 percent.

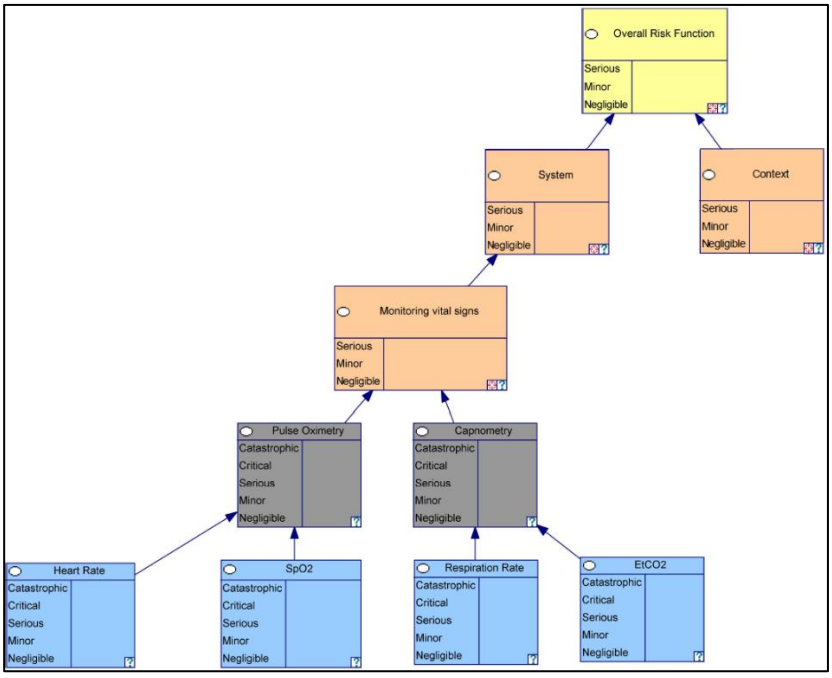


Figure 29 Construction of the NPTs of the monitoring nodes. Observe that either the values of the monitoring nodes or those of the aggregation nodes are discretized in a five-class classification defined by medical experts.

For the observable nodes such as the blue "SpO2" node, we also classified the whole data range measured by the sensors according to a five-class risk classification (derived from the EN ISO 14971:2012 severity classes). For example, for blood oxygen saturation (SpO2), the range of values is from 0% to 100%. The medical experts assigned the value ranges according to the risk of respiration depression. In the Table 3, we present examples of the SpO2 and EtCO2 value range classification.

Vital sign	**Severity**				
	Negligible	***Minor***	***Serious***	***Critical***	***Catastrophic***
SpO_2 (%)	≥ 94 %	91 %-94 %	85 %-90 %	≥ 75 % & < 85 %	< 75 %

Table 3 Example of five-class risk classification ranges.

Finally, we need to refine the Monitoring vital signs node so that it can precisely model the assessment from the vital signs. We implemented an adapted OR-function for the BN because the Capnometry node must be defined as the most relevant one in the relation, so the weight of the values in the CPT must represent this relation. We empirically defined this relation; the

Capnometry node has a value 68% higher in relevance compared to the pulse oximetry node.

Refining monitoring data nodes

Following the metamodel sequence, we need to specify the impact of the functional safety analysis in the risk metric. This requirement implies the challenging task of analyzing all the potential medical devices and systems that might take part in the MCPSoS. A straightforward solution is to limit the devices that can be integrated into the system; for instance, by defining a set of specific brands or homologated devices. This solution confines the MCPSoS to a controlled system with a low level of interoperability and threatens the definition of MCPSoS regarding the openness requirement. Another common solution is to deal with standardized devices and then allow a wide range of systems; however, defining a set of operational requirements for the system based on worst-case scenarios might seriously compromise the system's performance. In this work, we assume the state-of-the-art approach called runtime safety certification, which defines formalization of the guarantees provided by a system to automatically build up a system certification based on the active configuration. In this way, based on the provided guarantees defined in the current system's safety certificate, the system operation will deliver only the corresponding behavior.

To model these parameters in the risk metric, we established a set of guarantees for the devices that might take part in the MCPSoS. Such guarantees can be defined by the integrity level provided by the systems and are classified as value, timing, and delay. Next, we defined the uncertainty model for every single type of service guarantee. For example, pulse oximeter sensors might provide different levels of integrity as a result of different methods for collecting vital sign data or different brands / models of the product, etc. Hence, the update frequency impacts the provision and time requirements as well as the accuracy pose requirements for the value guarantees.

To define monitoring data nodes, we need to first identify the uncertainty model based on the provided system guarantees.

Refining monitoring data nodes requires further understanding of all monitoring devices/services (as specified in the metamodel's system functional realization layer) and their respective relations with the functional safety analysis layer. In other words, we already know what should be monitored, as this is defined by the domain experts at the level of the System in its Usage Context, but now it is necessary to define how the monitoring will be performed. Hence, we defined a set of tasks for further modeling such relations in the risk metric, as follows:

1. Tasks for defining sensor values nodes (see activity 2.5.1 in Figure 34)

2. Tasks for defining a node for the sensor guarantees (see the activity 2.5.2 in Figure 34)

3. Tasks for defining fuzzification strategy (see the activity 2.5.3 in Figure 34)

After reviewing all the monitoring nodes and the corresponding discretization values for the probability tables, we must specify all nodes in the risk metric that represent the raw data read directly from the sensors. We need to define different nodes for that in order to keep the responsibilities of the nodes clear and separate from further processing and classification, which we will define later on. Moreover, this is an opportunity to make sure that all sensor device values can be represented by the selected implementation model. In our application example, we defined a BN value node for representing such values. Note that in Figure 30 (a), the dark gray node at the bottom, the node "SpO2 sensor data value", represents the raw data read from the sensor and the node "SpO2" (blue node) is the node we already defined and discreticized for classifying the data.

Furthermore, we need to define how the guarantees defined in the metamodel will impact on the uncertainty of the sensor data. First, we need to identify guarantees and analyze them in terms of accuracy and availability, as was specified in the metamodel. We classified them based on the integrity level model, as this is a widespread and clear concept from standards. Safety engineers can define classes for the integrity level based on the provided

services guarantees, for example, using letters from A to D for the categorization from the lower guarantees to the most constrained guarantees. In this way, a blood saturation service that provides Guarantee_A would have a lower rate of availability and accuracy than a provided service classified with Guarantee_D, which should fulfill the highest requirements for both quality attributes. We assume that every system/service should expose its guarantees in a formalized way by means of a formalized framework such as we can see in techniques based on runtime safety certification [167, 196, 204]. In this sense, once every system/service exposes its guarantee data, we can derive the uncertainty impact of the raw data from the sensors on the rest of the risk metric.

In Figure 30, we show the further refinement for the "SpO2" data node. The nodes "SpO2 Distribution" and "SpO2_safety_Guarantees" are worth noting. The former is responsible for modeling an uncertainty level for the raw data read in the node "SpO2 Sensor value". We defined a Normal distribution [189, 215] whose mean parameter (μ) is given by the node "SpO2 Sensor value" and whose variance (σ^2) was empirically defined according to the current service guarantee as recommended by [4, 42, 140, 247]. The variance is inversely proportional to the guarantee, so the higher the guarantee, the lower the variance. Thus, we assure that higher guarantees will deliver lower uncertainty levels for the risk metric as well as more precise values to be considered in the risk metric. In Figure 30 (a) and (b), we show the difference in the uncertainty level propagated by the "SpO2 Distribution" node. In the first example, we can see a service-provided Guarantee D being applied to the raw data read as 95% for SpO2. It is worth noting that the resulting distribution variance spreads the values between 91% and 99%, which is classified by the node SpO2 as Negligible 84% and Minor 16%. In the In Figure 30 (b) , the service-provided Guarantee A spreads the values over a wider range of values (60% and 130%), which increases the uncertainty level of the SpO2 node, but still keeps the negligible classification, with the highest value being 56%.

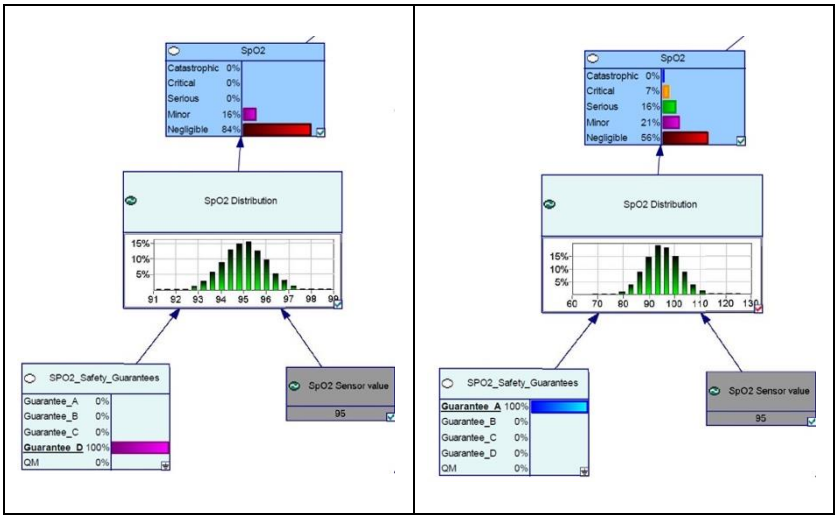


Figure 30: BN representation of the guarantees, distribution, and sensor data nodes for a heart rate sensor.

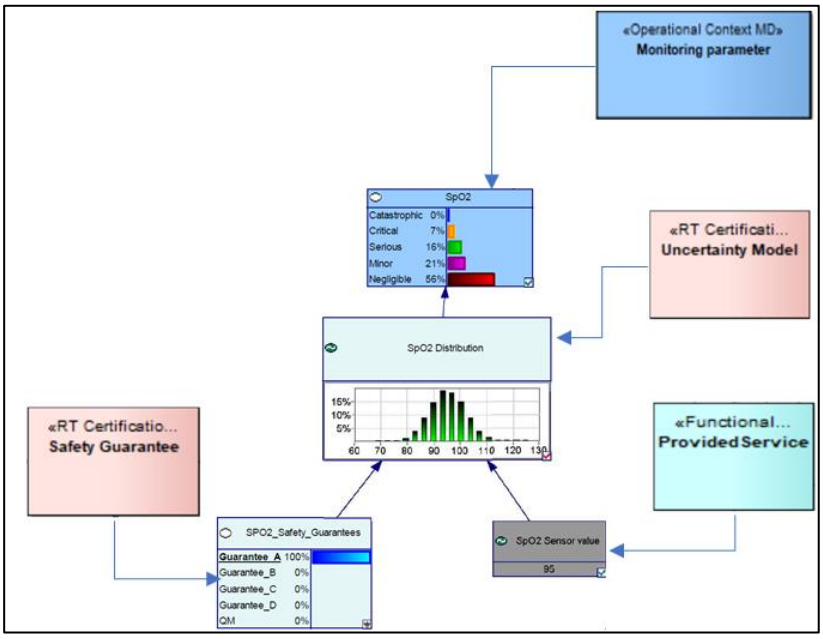


Figure 31 Piece of the Bayesian network generated by the proposed approach. We show the relation between the metamodel class elements defining (blue arrows) their respective nodes in the BBN.

Defining the system actuation data node

Medical cyber-physical systems change the environment based on reasoning about the monitoring data. Sensors and actuators cooperate in order to deliver some functionality at runtime. In this

sense, data from both system types are fundamental for situational risk assessment and must be considered a relevant part of the risk metric. For example, in the safety interlock function for patient-controlled analgesia, the infusion pump plays the fundamental actuation role of delivering the opioid to the patient. Hence, the set of parameters related to the infusion process does indeed matter to the overall risk assessment of the treatment.

We therefore need to define how the system actuation aspects will impact the risk metric, as we did for the monitoring parameters.

Initially, we need to identify, with the support of the metamodel and the medical experts, all the relevant system actuation parameters related to the treatment. For the example, at the level of System in its Usage Context, we can identify that the infusion service is the main service required for the PCA treatment. The relevant parameters for the treatment are: (a) the relation between amount and criticality of the infused opioid; and (b) the programmed lockout time interval between doses. Furthermore, system and safety engineers specify the devices responsible for delivering this service at the System Functional Realization level, which are infusion pumps and the variety of infusion services they provide. As explained in Section 4.1, the safety interlock scenario can utilize different services provided by the infusion pumps, such as regular coordinated infusion, infusion controlled by timed token, and remote stop infusion command.

Finally, we need to identify the services' safety guarantees required/provided for the implementation of the safety interlock scenario in order to define the uncertainty model for the fuzzification of the data in the risk metric. For example, aspects related to the precision of the infused dosage or timing might vary according to the types and brands of infusion pumps. Particularly, for this example scenario, the medical experts and safety engineers defined as minimal required guarantees the highest integrity level for the services provided by the infusion pumps that might be integrated into the overall system. Hence, we defined a three-degree scale for the identified parameters without considering any further noise caused by the different services' uncertainty models.

To implement all the impacts of the actuation parameters in the risk metric, we need to place and structure the newly identified nodes in the current tree as well as to define and refine the node probability tables. In Figure 32, we show how we implemented these parameters in the risk metric for PCA treatment.

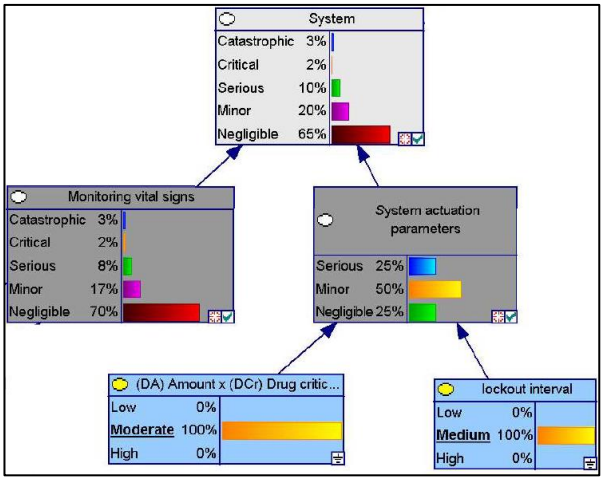


Figure 32 Piece of the Bayesian network generated by the tasks of the refinement of the actuation node.

We defined the System Actuation Parameters node for aggregating the impact of all the actuation parameters on the risk metric. In Figure 32, the System node aggregates the evaluated probability from both the Actuation node and the Monitoring Vital Signs node. Therefore, its NPT needs to be updated to better express the probability of the system status to the overall situation assessment. For this example, medical experts and safety engineers decided to make the Monitoring Vital Signs node more relevant to the risk metric than the System Actuation Parameters node.

It is worth noting in Figure 32 that the CPTs of the System Actuation Parameters node were implemented using three-level parameters. The (DA) Amount x (DCr) Criticality node and the Lockout Interval node are discreticized using a low, medium, and high scale for representing the increasing criticality levels as can be seen in their parent node. Finally, the System node needs to aggregate the estimated probability of both children nodes in its node probability table, which we defined empirically with the help of the medical experts' input. For this table, we specified

higher weights for the Monitoring Vital Signs node than to the System Actuation Parameters node due to the former's direct relevance for the identification of the patient's situation.

4.5.3 Refinement of the context node

The Context node defines a precise overview of the complete situation environment for the patient's treatment. As described in Sections 4.1 and 4.2, it brings light to relevant aspects related to the treatment procedure, additional details, and the relevant patient profile.

These elements enhance the accuracy of the situational risk assessment by specifying elements of the System in its Usage Context. According to the meta-model, we need to define: (a) any relevant treatment procedure performed by the caregiver in the treatment; (b) additional treatment details for the patient; and (c) the relevant patient history for the risk assessment. The update frequency of vital signs is dependent on the type of prescribed treatment. The frequency can be set according to the suggestions of physicians or best practice medicine guidelines.

These information elements are relevant for several other treatments where aspects related to the caregiver can define the risk of the situation. For example, the position of the x-ray operator during a radiography, or the awareness level of the caregivers in the intensive care unit. For the current example scenario of patient-controlled analgesia treatment, we interviewed the medical experts and they did not define any relevant aspects related to the treatment procedure. However, a further improvement in this scenario example could include the awareness level, experience, or education level of the caregivers for the risk metric.

For the current example, we identified relevant elements for additional treatment nodes, such as oxygen supplementation and relevant medication, that might impact the opioid's effect on the patient. Hence, we built a structure for the context based on the refined nodes. As can be seen in Figure 33, we defined two nodes for refining additional treatment aspects, namely "Taking Other Medications" and "O2 Supplementation" (these nodes are

depicted as bar charts). The former defines only a CPT based on the observation of relevant medications such as “yes” or “no”. The state of the patient taking relevant medication increases the criticality of the whole context. The O2 Supplementation node defines three levels of oxygen supplementation; the higher the supplementation, the higher the risk level for the patient.

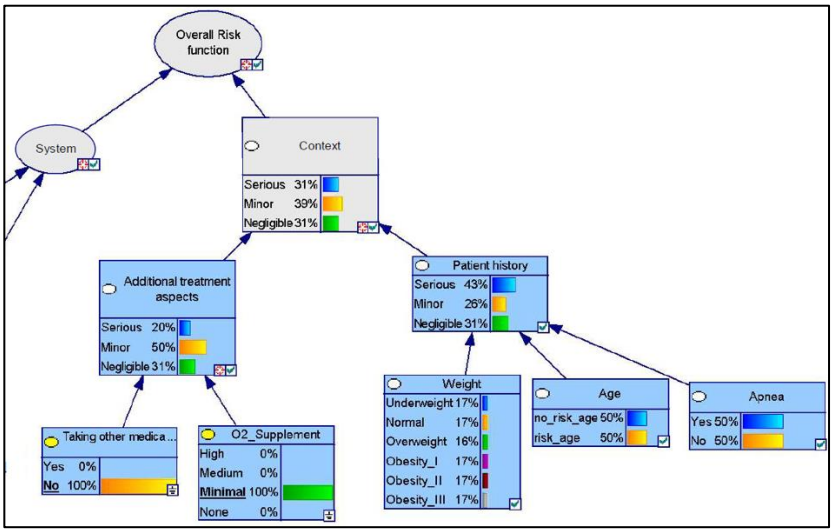


Figure 33 Part of the Bayesian network refining the context elements (additional treatment and patient history)

Furthermore, we also refined the Patient History node with relevant information for characterizing the patient situation for PCA treatment. Medical experts defined weight, age, and apnea as children nodes. In Figure 33, we show at the bottom right the apnea node CPT with two options, “yes” and “no”. In the case of “yes”, the criticality level in the Patient History node increases. Regarding the Age node, if the patient is older than a defined risk age, the risk level increases as well. For the Weight node, we classified five scales from underweight to obesity level III. The more obese a patient is considered, the higher the deemed risk level for the situation. It is also important to observe that the Context node has the power to invalidate the PCA treatment in case several conditions are defined for the patient. For the Context node, we decided to keep a three-level probability table (negligible, minor, and serious) in order to simplify the CPT in the overall risk function and to better represent the semantic

meaning of the node in the whole risk metric. This node therefore aggregates the probabilistic risk for the relevant treatment conditions and its probability table must be empirically refined in order to represent the medical experts' demands regarding details related to the patient's context.

Following the meta-model, we also have to define how the context data will be gathered and transmitted to the risk metric through the System Functional Realization node. Currently, most healthcare providers have information systems for managing complex medical procedures and health data processing [16, 223]. We can hence assume that a hospital information system can be used to collect the context data from patients, treatments, and caregivers so that the risk metric can be fed with the context information to perform situation assessment. Hence, for the sake of practicality, we did not consider functional safety analysis in order to avoid any additional complexity for such systems due to the higher requirements of the safety engineering process.

4.5.4 Definition of the defuzzification strategy

Defuzzification is the process of converting a fuzzified output into a single crisp value with respect to a fuzzy set. The defuzzified value in a FLC (Fuzzy Logic Controller) represents the action to be taken in the controlling process. In [189], various defuzzification strategies are presented that can be applied depending on the situation at hand. We selected the weighted average method for defuzzifying the values from the top node (in the example: the Overall Risk function). Furthermore, the evaluation module in the safety monitor (for more details, see Chapter 5) considers this crisp value for the runtime evaluation algorithm to determine whether the risk value is acceptable or needs to be controlled. Hence, the defuzzified value is defined as:

$$x^* = \frac{\sum \mu(x).x}{\sum \mu(x)} \tag{4}$$

Where $\sum$ denotes the algebraic summation of the valued states $\mu(x)$ and x is the element with the maximum membership function. In the next section, we will detail how to adjust the maximum membership function x according to the system configuration in order to increase the risk faster for less reliable

abstract configurations and make it more stable for more reliable abstract configurations.

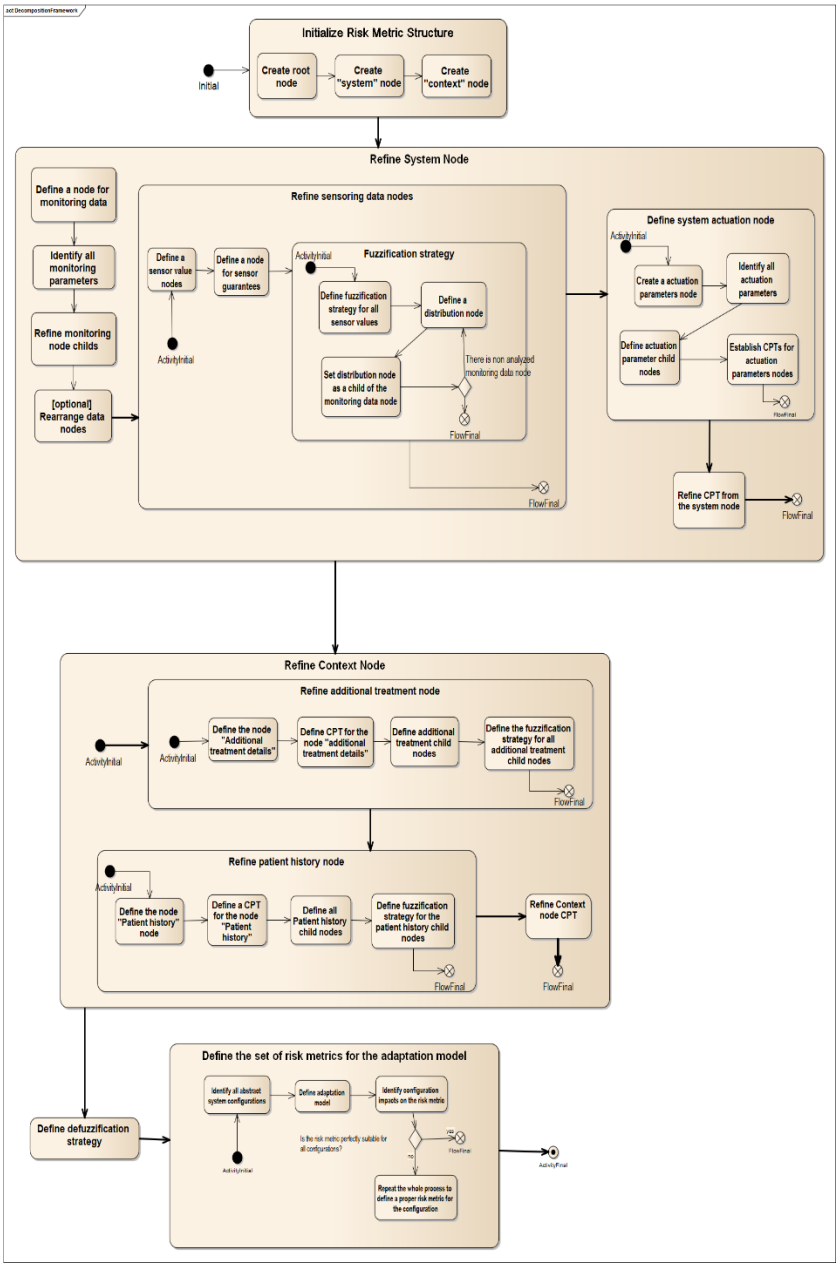


Figure 34 Complete view of the approach for deriving the dynamic risk assessment model

4.6 Definition of the adaptive risk assessment model

Runtime system adaptations of cooperative MCPSoS constantly occur during the ward routine in hospitals. Once the medical team has defined the treatment and the medical devices required for supporting the procedure according to the standard protocols and hospital policies, the patient is settled in a monitored bed. However, several problems might happen during long procedures, such as:

- Patient moving and disconnecting attached devices
- Patient deliberately deciding to disconnect devices
- Devices eventually experiencing malfunctions
- Incompatibility between devices and the hospital information system
- Limited set of devices for all the patients in treatment

For these reasons, different runtime system configurations can be activated at runtime during the treatment. This characteristic poses to the risk metric the challenge of having to be adaptive in order to deal with system changes and still keep up the risk assessment. Moreover, the risk metric must reason on whether such changes might endanger the risk assessment and/or the overall treatment itself.

In the previous section, we derived a very complete risk metric for patient-controlled analgesia treatment based on the proposed meta-model and guidelines. We assumed that all the monitoring devices would always be connected to the system with high availability and no adaptations could interfere with the risk metric. However, in the case of a simple adaptation, for instance a respiration monitor disconnection, the risk metric will provide wrong values for EtCO2 and respiration rate data. This can therefore compromise the overall risk assessment and then make the PCA care unavailable or lead to high risk levels for the treatment.

As presented in the state-of-the-art section, the current approaches consider vital signs for risk assessment; however, they do not provide any support for system adaptations. The risk metric intrinsically depends on the abstract system configuration which it was designed to operate. We therefore present in this section an approach for improving the risk assessment to include considering system adaptations.

4.6.1 Definition of an adaptive inference model

The adaptive inference model relies on a structure based on automata for managing the risk metrics according to the active abstract configuration. The presented framework is based on a Bayesian Network, which provides a solid mathematical framework for runtime probabilistic risk assessment. Furthermore, we combined the risk metrics with the automata definition in order to provide a clear idea of the transition between the configurations. We therefore defined the Dynamic-BN framework for identifying all the needed elements and then specifying transitions and actions that need to be taken when the considered adaptations occur.

Formal definition:

Dynamic-BN can be defined as a tuple **<C, C_0, G, B, I, T>:**

1. **C** is the set of **abstract configurations** $C = \{c_1, c_2, \dots, c_n\}$, where ci is an abstract configuration that the system might assume at runtime. An abstract configuration is defined by a set of orchestrated devices provided by the corresponding devices.

2. C_0 is the **initial abstract configuration** $C_0 = \{c_0\}$, where c_0 is the label of the state for the initial abstract configuration.

3. **G** is the **set of guards** $N = \{g1(C), g2(C), \cdots\ gn(C)\}$, where $gi(C)$ is the Boolean expression defined on C, where it checks whether some adaptation has occurred in the system of systems.

4. **B** is the set of **Bayesian networks** $B = \{bn_1, bn_2, \cdots bn_n\}$,where bn_i is defined as in the section BN; and i corresponds to the number of configurations. Hence, every single abstract configuration defines a BN for performing risk inferences.

5. **I** is the set of **inferences** I = $\{i_1, i_2, \ldots, i_n\}$, where i_j is the inference of B.

6. **T** is a **set of transitions** $T = \{t_1, t_2, \cdots t_n\}$, where ti is the transition between two configurations defined on a guard and an inference, as $t_i \subseteq C \times G \times I \times C$. It is the formal representation of a system adaptation.

Every abstract configuration can be defined as a tuple $< S, M, S_0, G, T >$, which defines its adaptation plan at the service level:

1. **S** is the **set of services** $S = \{s_1, s_2, \ldots, s_n\}$, where si is any potential service that the corresponding configuration provides or requires to perform its function. An abstract configuration can be further defined as a set of orchestrated services required and provided by the respective integrated devices that cooperate to provide a joint nominal service.

2. **M** is the **minimal set of services** $M = \{s_1, s_2, \ldots, s_m\}$, where sj is the minimal set of services that defines a corresponding abstract configuration. If these minimally required services are not active in the current configuration, the system will adapt to another configuration that is fulfilled by the currently active services; in case no configuration can be activated, then the system will activate the state Configuration – no operation.

3. $\boldsymbol{S_0}$ is the initial **set of services** $S_0 = \{s_0\}$, where s_0 is the system running state that defines the initial set of services so that the corresponding abstract configuration can always be activated.

4. **G** is the set of **guards** $N = \{g1(C), g2(C), \cdots gn(C)\}$, where $gi(C)$ is the Boolean expression defined on C, where it checks whether some adaptation has occurred in the system of systems. In this sense, we consider adaptation as a change in the set of services provided or required by the system.

5. **T** is a **set of transitions** $T = \{t_1, t_2, \cdots t_n\}$, where ti is the transition between two sets of services defined on a guard, as $t_i \subseteq C \times G \times C$. It is the formal representation of a system adaptation.

Deriving an adaptive risk metric based on Dynamic-BN

To derive a Dynamic-BN for risk metric management, it is necessary to arrange all the relevant information regarding the adaptations and abstract configurations.

We first need to identify the abstract configuration model that determines how the system will adapt between the configurations. The system adaptation plan also defines with how much autonomy the adaptations will occur and what the requirements and the minimal architecture for performing the designed system functions are. Such information is usually specified at design time by the system architects.

Predicting all system adaptations is practically impossible due to the wide range of potential devices that might be integrated into the system during the whole lifecycle. Some medical devices might not even have been created yet at the time the system project is designed. Moreover, different types of devices can provide the required services. For example, the blood saturation data service can be provided, with different confidence levels, by multiparameter cardiac monitors or by pulse oximeters. Therefore, we can define abstract configurations based on the required services provided by the devices.

The process starts with the definition of abstract configurations and is then further refined to the services required for the architecture. In Figure 35, we show an automata model that represents the abstract configurations and the adaptation plan. We

can thus identify the required elements according to the formal definition of the Dynamic-BN, as follows:

1. Abstract Configuration Set: {Configuration 0 – no configuration; Configuration 1; Configuration 2; Configuration 3}

2. Initial Configuration: {Configuration 0 – no configuration}

3. Set of Guards: {[Pulse Oximetry is integrated into the system]; [Disconnection of Pulse Oximetry from the System]; [RespirationMonitoring is integrated into the System];…}

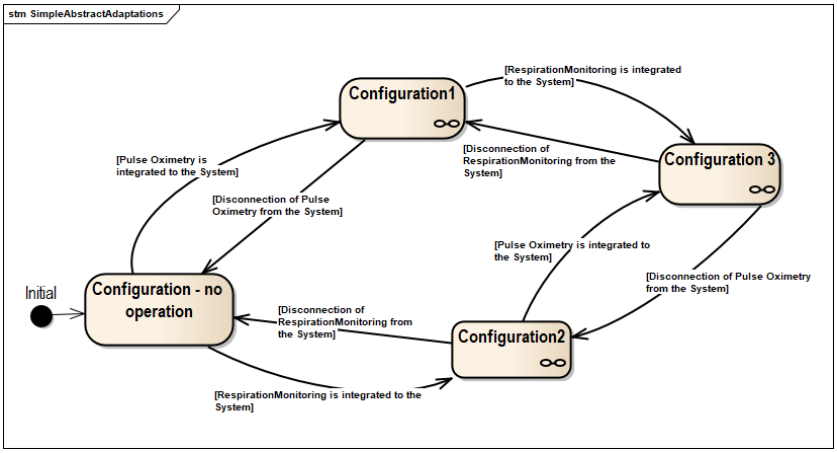


Figure 35: Adaptation model defining the risk metrics for each abstract configuration.

1. Set of Transitions: {[“Configuration – no operation” to Configuration 1 when the Pulse Oximetry is integrated into the System];…}

It is worth noting that the specifications for required devices are not constrained to particular devices; rather, we focused on the types of devices (pulse oximetry or respiration monitoring) that can perform the required tasks/services. Moreover, we further refined the requirements and services orchestration needs for every abstract configuration, as depicted in Figure 36.

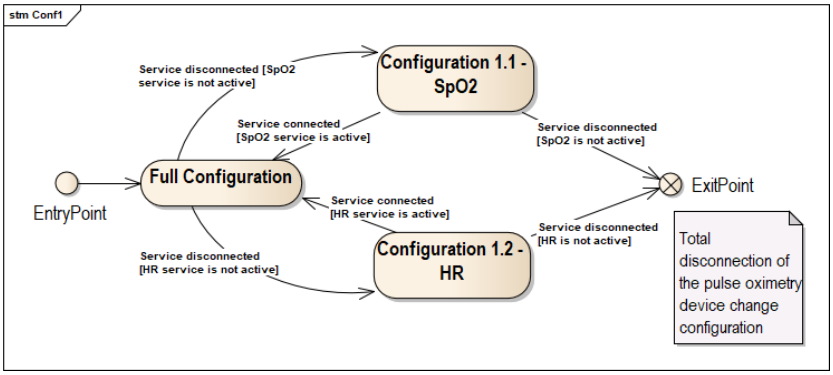


Figure 36: Service adaptation plan for abstract configuration 1

Within every abstract configuration, it must be specified how the services orchestration shall work and what the acceptable adaptations and minimum service requirements for the configuration are to be. In Figure 36, it can be seen that the entry point for the automata is the full configuration for the abstract configuration, which means both services related to pulse oximetry (blood saturation and heart rate data). After that, a set of transitions and guards are specified to allow at least one service to be available for the abstract configuration; in case both services are unavailable, the exit point specifies that this abstract configuration should be left. As we can see in Figure 35, the exit point represents the transition from Configuration1 to Configuration 0 – no operation.

After defining the adaptation plan, we need to specify how the risk metric will perform the dynamic risk assessment for the system given the set of abstract configurations that the system might activate. First of all, we need to analyze how the adaptations and abstract configurations will impact on the risk metric. Furthermore, the implementation technology utilized for realizing the risk metric might matter in this step. For example, Bayesian Networks struggle with runtime changes in their NPT and their structure. BN-derived models such as Oriented Object BN or Dynamic-BN support some changes in the structure as well as small adaptations and inferences; however, all of them have shown limitations when a change in the network structure requires a deep redesign of several other nodes in the model [28, 34, 58, 105].

Hence, we decided to specify Dynamic-BN by integrating a Bayesian Network for every single abstract configuration (as can be seen in the formal definition above). These models need to be managed by a system monitor (defined in the next chapter) and need to be activated as soon as a corresponding abstract configuration is identified and running. In this way, every active abstract configuration also brings its own BN, so that the most suitable inference can be performed by the system. This feature enables higher inference precision and flexibility for defining BNs based on the required set of devices/services.

Following the Dynamic-BN specification, we defined three Bayesian Networks referring to the abstract configurations of our example, namely: RiskMetric1 (for Configuration 1 – pulse oximetry monitoring), see Figure 37; RiskMetric2 (for Configuration 2 – respiration monitoring), see Figure 38; and RiskMetric3 (for Configuration 3 – both sensors), see Figure 39.

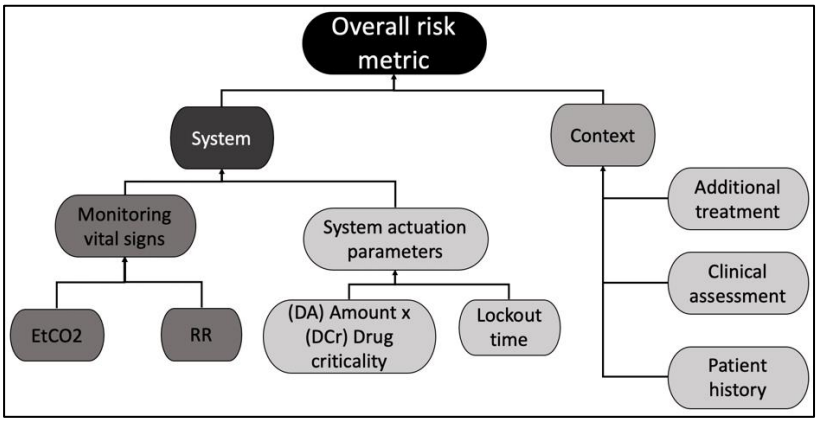


Figure 37: Bayesian Network model defined for the risk metric of abstract Configuration 2, where sensoring is performed only by the respiration monitor.

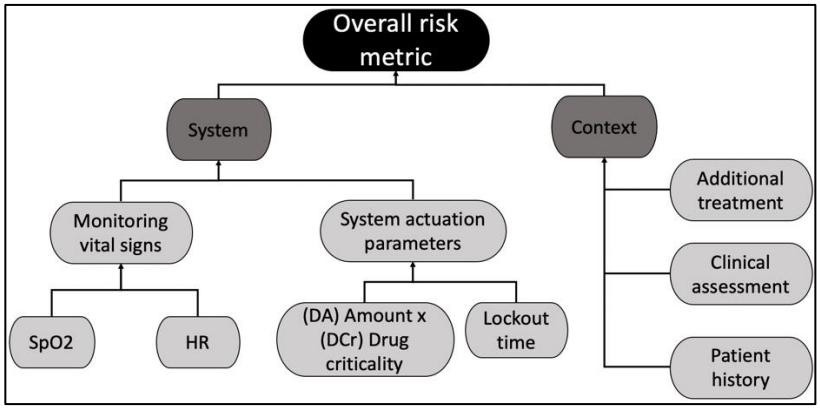


Figure 38 Bayesian Network model specified for the risk metric of abstract Configuration 1, where sensoring is performed by a pulse oximetry device.

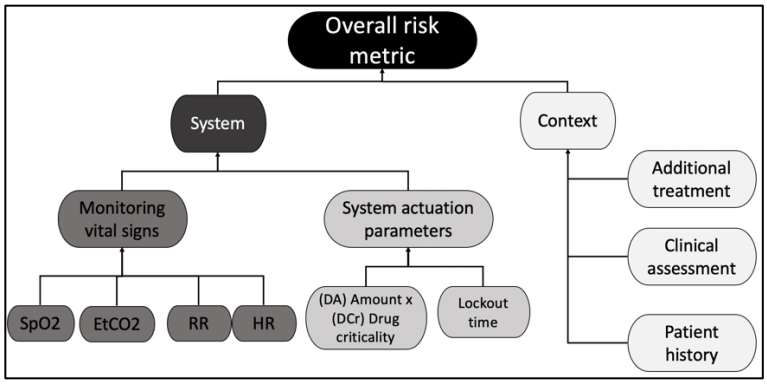


Figure 39 Bayesian Network model specified for the risk metric of abstract Configuration 3, where sensoring is performed by both the pulse oximetry and respiration monitoring devices.

Every single risk metric has its own structure, NPT particularities, and demands on the system. In the three figures above (Figure 37, Figure 38, and Figure 39), the System and the Context nodes are shown with different colors. Figure 38 shows that the abstract configuration is less reliable for patient-controlled analgesia risk monitoring. Therefore, we defined the NPT of the Overall Risk Metric node to give the same weight to the Runtime Monitoring and the Context nodes, as shown by the same color used for the nodes in the picture. However, in Figure 39, the System node is shown with a dark color, which represents that this node is more relevant than the Context node. Likewise, a similar relation between the Monitoring Vital Signs node and the System Actuation Parameters node was established for the System node,

considering the monitoring to have more relevance. Hence, as can be seen besides the network's structure, the relevance implemented in the node probability tables changes according to the currently activated abstract configuration.

In order to derive the risk metric management plan, it is necessary to combine the adaptation plan (as defined in Figure 35) with the respective risk metrics. In Figure 40, we depict how the transition model should work for the identified system configurations of the case study. The blue arrows in the figure indicate how the system adaptations activate the corresponding system configuration and their respective risk metric. For instance, if a running configuration is monitoring the patient with pulse oximeter and capnometer (Risk Metric 3) and someone disconnects the pulse oximeter, the system monitor needs to identify that there was an adaptation and instantiate the proper risk metric, in this case activating Risk Metric 2. Thereupon, the risk shall be assessed based on the particularities (defined by CPTs, defuzzification approach, etc.) of its proper risk metric.

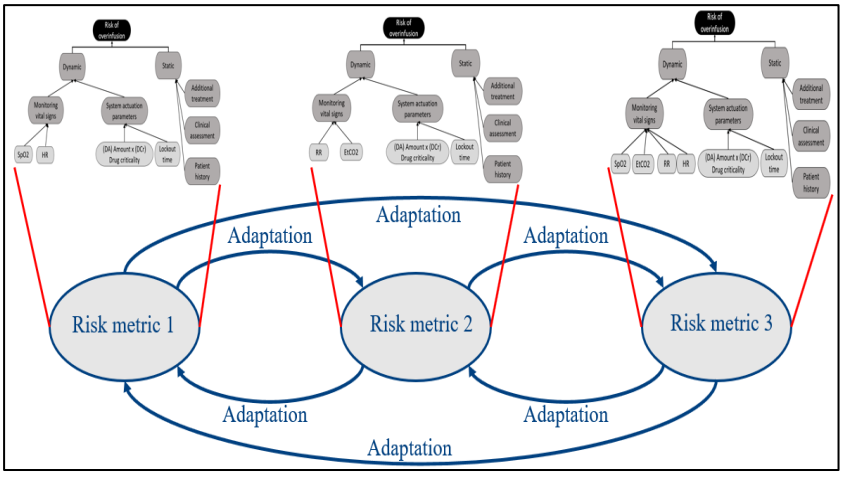


Figure 40 Risk metric management for the adaptation model of the case study.

5 An approach for enabling runtime risk classification and risk control specification

In this chapter, we detail our approach for enabling runtime risk evaluation, which aims to support cooperative medical CPS in determining whether the situational risk is acceptable or tolerable for performing the current system function.

Moreover, we detail in this chapter a technique for defining risk control based on the proposed risk evaluation framework. This strategy aims to deploy and specify countermeasures that the system must activate in order to keep the risk at an acceptable or tolerable level for performing the system joint function.

This corresponds to the second contribution of this thesis, as specified in Chapter 3, that makes up our approach to dynamic risk assessment of cooperative medical cyber-physical systems of systems. Observe in the Figure 41 the red square over the box "2. Runtime Risk Classification" which highlight the referred contributions of this chapter.

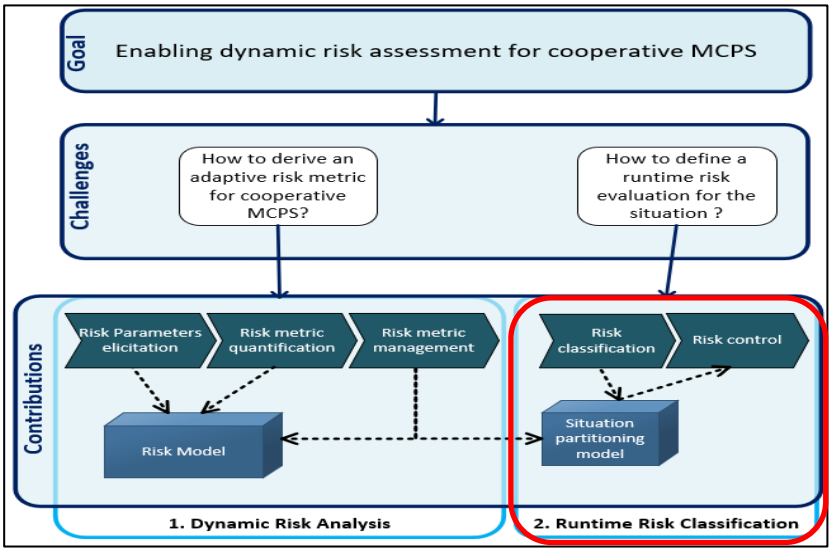


Figure 41: Contribution 2: Runtime risk evaluation and control.

Therefore, we organized this chapter according to the two main constituents (runtime risk classification and control) as follows: in the section 5.1 we present the main drivers for the specification of runtime risk classification and countermeasures; in section 5.2, we present the runtime risk classification technique; in section 5.3, we show the risk situation analysis framework for supporting the definition of dynamic models for risk classification at runtime; in section 5.4, it is shown the approach for allocating the risk countermeasures according to the current risk situation; finally, in the section 5.5, we present an extended reference architecture model for allocation of the system functions responsible for risk management (analysis, evaluation and control).

5.1 Drivers for runtime risk classification and risk countermeasures in MCPSoS

As previously mentioned, the core motivation for creating the approach for runtime risk evaluation presented in this chapter was to support the situational risk analysis at runtime for dynamic systems such as cooperative MCPSoS. Since we have derived an approach for performing risk assessment at runtime, risk evaluation and control need to be integrated with this strategy in order to deal with all the challenges addressed in this thesis. We considered the three main drivers that guided the creation of the approach: (i) addressing conflicting requirements of safety and availability; (ii) inherent system dynamicity; and (iii) complying with best practices as described in the recommendations of standards.

5.1.1 Addressing conflicting requirements of safety and availability

Safety engineers must often deal with divergent requirements regarding safety, availability, performance, and so on. These conflicts tend to increase due to the dynamicity of MCPS. For example, the innumerous combinations of context and system configurations make it impractical to perform design time analysis of the proper treatment and system set for the infinity of patients' statuses and hospital environments. Therefore, worst-case assumptions are made for the definition of system safety assurance and thus as compromise between performance and availability [132, 231, 233].

In this context, risk classification and risk control measures must take into account that system availability enables the treatment of patients. However, when the system constrains the functions, patients will not be able to use the treatment provided by the system. For example, low system availability for analgesia can make pain relief treatment impracticable. Hence, countermeasures must be allocated well according to the risk level in order to avoid excessive function restrictions while preserving an acceptable risk level.

5.1.2 Inherent system dynamicity

Considering that MCPSoS are open to cooperation and reconfiguration of their architecture at runtime, risk classification demands a very precise risk assessment in order to accurately determine the level of acceptance.

Moreover, eventual reconfigurations or system adaptations can indirectly affect the risk classification since the risk assessment might also change depending on the system configuration. For example, the risk monitor needs to consider the fact of a system reconfiguration, and/or a contextual change can make the system risk unacceptable.

5.1.3 Complying with best practices described in the recommendations of standards

A reasonable requirement for the approach is that safety engineers must define an acceptance criterion to enable runtime risk classification. A recurring blueprint from the state of the practice and specialized literature is the adoption of an acceptability matrix for determining acceptance criteria for safety-critical systems [90, 91, 95, 254]. This approach supports the definition of a relation between quantitative frequency data and qualitative severity analysis for each hazardous situation. This relation defines the risk of the situation for the analysis. Then safety engineers establish a threshold area in the table for acceptable risk in order to define which situations and conditions exhibit acceptable risk. Nevertheless, this activity is even challenging for non-adaptive systems due to the completeness requirement for the identification and specification of operational situations [112].

Open and adaptive systems make this requirement practically unattainable due to their dynamicity.

Therefore, a risk classification approach must assume all the requirements from the standards and support safety engineers in defining acceptance criteria for allocating effective countermeasures.

5.2 Overview of the runtime risk classification and control approach

The main aim of the runtime risk classification and control approach for cooperative MCPS is to evaluate the current risk for a given situation/clinical scenario. This makes it possible to continuously check the matching between the current top-level safety system guarantees of the MCPS (which might also be subject to dynamic change due to, for instance, an added sensor) on the one hand and the current top-level safety requirements on the other. Based on this, a sufficient level of safety can be ensured while performance and availability can be optimized at the same time.

However, system dynamicity adds an extra load of complexity due to various reasons: runtime adaptations that might change the system capabilities; some level of autonomy of the performed operations; and relevant changes in the context. Such characteristics can affect either system actuations, which might limit/enable some countermeasures or system sensors, which can restrict the system's awareness level. This directly affects the current risk because the system variables define the risk according to our framework. Moreover, ALARP principles [78, 91] define a risk evaluation strategy for a system under development, which is not the case in runtime risk evaluation for MCPS. Therefore, defining risk classes for situations must take into account that evaluation and risk control must be conducted autonomously at runtime.

The runtime risk evaluation approach comprises two main steps: (i) situation analysis framework; and (ii) risk control deployment. The former is responsible for classifying the situational risk according to the previously defined dynamic assessment approach, specifying the transitions between the identified states

given system behavior and context dynamicity. The latter is responsible for guiding the allocation of countermeasures considering both the system modules responsible for monitoring and adapting the system behavior and the dynamic architecture according to the specified control actions.

5.3 The situation analysis framework

The situation analysis framework defines a formal concept of "situation" to support safety engineers in identifying and describing classes of situations that the system will need to use for evaluation at runtime. Hence, after the classes and relevant parameters that describe a situation have been defined and given the changes in these parameters, at runtime the system will assess the aggregated situational risk to classify it according to the previously defined classes.

To specify situations in the framework, we extended the formalization from [144, 162] to consider two new points. First, we also considered relevant context variables instead of only variables concerning the system state. Moreover, we considered an aggregate function to calculate the risk level (the previously presented risk assessment approach) for the classification of situations and the specification of state transitions.

Hence, we define a situation based on the following propositions:

- Let $x \in X$ be a tuple of safety-relevant variables, such that $x = < x(1), x(2), \dots x(n) >$ represents the set of discernible **system** states. For these system states, we consider not only variables that define specific fine system settings (such as infused amount of opioid or guarantees about a device's precision) but also the current active configuration with its sensors and actuators.

- Let $y \in Y$ be a tuple of safety-relevant variables, such that $y = < y(1), y(2), \dots y(m) >$ represents the set of discernible **context** states. For instance, we might consider all context variables defined during the elicitation phase, such as patient's blood pressure, blood saturation level, treatment details, etc.

- Let $s \in Sit(x, y)$ be a situation defined by a set of safety-relevant system and context system variables, such that $S = Sit(x, y) = X \times Y = \{(x, y) \mid x \in X \text{ and } y \in Y\}$.

- Assume the previously defined risk function as $Rf: S \rightarrow \mathbb{R}$. The function establishes a relation between the set of variables describing the situation and a real value representing the situational risk level.

- Assume the definition of the safety invariant from [144, 145] as *a sufficient condition for avoiding a hazardous situation*. Conservatively, in case of any violation of the safety invariant, there is no possible recovery action by the system. In our framework, we also define safety invariants by their unacceptable risk level. They can be represented by a true valuation of a function: $SI: S \rightarrow \mathbb{B}, i.e., SI(Sit(x, y)) = true$. Where *true* means that the described situation is assessed with an acceptable risk level and *false* means an unacceptable risk.

- Catastrophic situations can thus be defined by the set of situations deemed to have an unacceptable risk level: $Sit_{cata} = \{s \in S \mid \overline{SI(Sit(x, y))}\}$.

- Accordingly, we can define catastrophic situations by the following assertion: $Rf(s) > Accepatable$. This implies two clearly distinct classes: (a) the catastrophic situations class:
  $Sit_{cata} = \{s \in Sit \mid Rf(s) > acceptable\ risk\}$;
  and (b) the acceptable situations class:
  $Sit_{safe} = \{s \in Sit \mid Rf(s) \leq acceptable\ risk\}$.

Given the framework definitions, we have a clear idea of catastrophic and safe situations based on the assessed situational risk. In Figure 42, we show a graphic representation of the concepts defined above as well as some situation examples labeled as s_1, s_2, s_3, and s_4. Considering the set of safe situations, let us compare s_1 and s_2 according to their position in the set. Assume that the situation s_2 is assessed with a higher risk level than the situation s_1 due to the safety-relevant elements that describe each situation.

Moreover, the probability of s_2 crossing the acceptable risk threshold and becoming a catastrophic situation is higher than that of s_1. Applying ALARP [78, 91] principles for risk management, we should try to reduce the risk of the situation s_2 in order to decrease its probability of becoming a hazardous situation. Conversely, the situation s_1 does not have such a requirement due to its lower risk level. We must hence divide the situational space classes in order to apply different risk control actions to manage keeping the situations in the safe region.

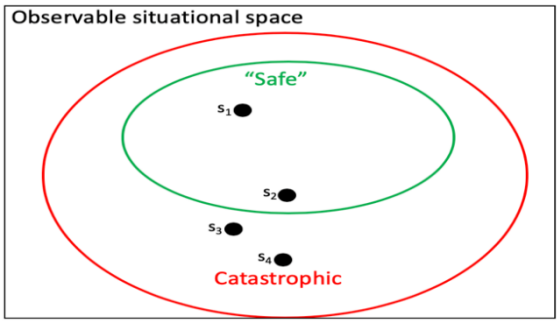


Figure 42: Situational space breakdown

We could further refine catastrophic subclasses (see the situations s_3 and s_4 in Figure 42), but the practical gains of performing such task are not relevant since we need to avoid reaching this state class due to the fact that the system's design is not prepared for mitigation. In the case of a situation classified as catastrophic, there are no system actions to reduce the risk level, so all possible actions should be taken to avoid catastrophic states.

5.3.1 Identifying warning states

In order to identify how many safe situational classes we should split into warning states, we can take the recommendations from the standards such the ALARP principle. It requires that situations with any risk shall be reduced so far as reasonably practicable. In this sense, the risk is classified into three regions: intolerable (it cannot be justified except in extraordinary circumstances); tolerable (it is undertaken only if a benefit is desired); and broadly

acceptable (no need for further measures to reduce it). Hence, within the tolerable region, the activity is allowed to take place, assuming that the associated risks have been made as low as reasonably practicable by the designed risk control measures.

It is worth highlighting the difference between tolerable and acceptable risk. Tolerability indicates a willingness to operate the system functions to obtain certain benefits. Acceptability defines the minimal risk requirement for performing the system functions without causing any harm. Therefore, the definition of unacceptable risk is often used for specifying catastrophic situations. Intolerable risk, on the other hand, means that the risk associated with the system operation is not justifiable given the benefits, and although the hazardous event has not happened yet, it is imminent.

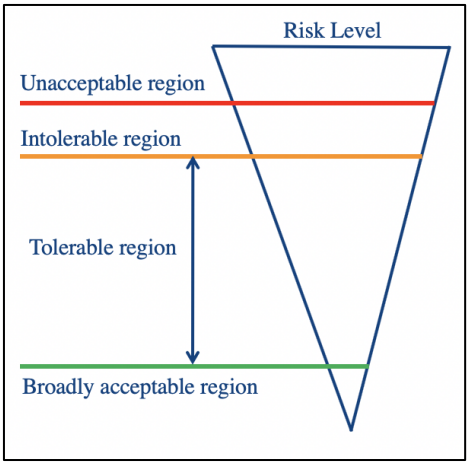


Figure 43: Risk regions in ALARP

In Figure 43, we illustrate the risk regions defined by ALARP, with the addition of the unacceptable region, which obviously represents a higher risk level compared to the intolerable region. Observe that the distance between intolerable and unacceptable may vary according to the system operation, the safety assurance level, and the catastrophic situations. It is also worth noting that the tolerable region is represented as the biggest area, as keeping the risk low (broadly acceptable

region) may impose excessive operational costs or may undermine system availability.

Theoretically, there are uncountable sets of subclasses. However, this division is a practical option for deploying system requirements for risk control actions. The definition of classes must thus keep the situation's risk under control and promote reasonable system availability, as previously defined in Subsection 5.1. Hence, we extended the ALARP principles to define the situational risk classes depicted in Figure 44.

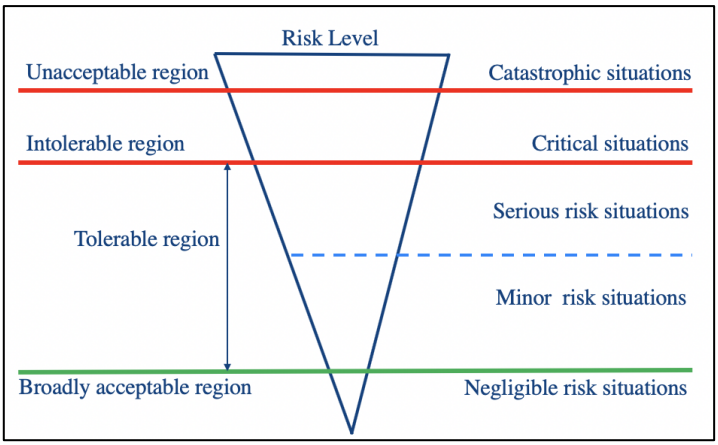


Figure 44: Definition of risk classes

In Figure 44, the defined risk classes represent situations ranging from unacceptable high-risk situations to broadly acceptable risk situations. We further detail the concept of each class below:

1. ***Catastrophic situations*** were already defined in the previous section This class characterizes a hazardous event that can end up in harm for patients for instance, an overdosage event might cause harm to a patient such as respiration failure, permanent disability, or death. This risk level imposes no requirements for risk reduction because the system should never let the situational risk reach these situations.

2. ***Critical situations*** define a risk class that have an intolerable risk for the current operation. Given the

runtime usage context of MCPS, the risks are too high to justify the treatment benefits. At this risk level, the system should never be allowed to conduct the treatment and a medical expert should take over total control of the treatment. No risk reduction is required for this risk class since the system should avoid reaching this risk level region.

3. ***Serious risk situations*** define an undesirable (yet tolerable) risk level region with restrictive requirements for system operation. This risk class should define the last barriers for avoiding that the situational risk level reaches critical levels.

4. ***Minor risk situations*** comprise a set of situational risks that is tolerable with some operational restrictions. For these situations, requirements to reduce the risk would impose severe restrictions on system availability and, consequently, impact the treatment's availability for the patients. For instance, limiting opioid infusion for PCA treatment would decrease the overdosage risk, but would increase the patient's pain level due to a lower rate of infusion.

5. ***Negligible risk situations*** define the desirable risk level for system operations. Here, no operational restrictions or requirements for risk reduction are imposed. At this level, the patient is allowed to have full availability of the treatment given the lower risk level of these situations.

The risk classes support a clear understanding of the current risk that each situation might assume over time. These risk classes are not a steady set; some applications might require a less detailed arrangement. For example, in the presented case study, we decided to establish four risk classes by combining the critical and catastrophic classes into one, since the practical effects for the allocation of countermeasures are similar. The decision about how many risk classes are suitable for the application should take into account diverse aspects, such as:

- The type of countermeasures that can be taken to reduce the risk. We must consider the establishment of safety barriers to avoid or prevent a hazardous event from occurring; control (something that maintains a condition or preserves a controlled system state); or mitigation of consequences if the unwanted event has occurred. For MCPS, for instance, mitigation measures are seldom defined in the system itself.

- The cost for risk reduction. As we have mentioned along of this thesis, availability will often conflict with safety. For example, a straightforward risk reduction measure for avoiding overdosage is to stop the infusion; however, this might increase the patient's pain (studies [85, 160] show that a painful treatment can increase the patient's stay in the hospital and the risk of getting other complications).

- The time in which the situation changes the risk. The faster the context and the system changes, the more agile the barriers need to be to reduce the risk.

- The system's autonomy level. Currently, we have found applications [72] only for safety interlocks. However, in the near future, applications will perform tasks with a higher level of autonomy, such as adjusting the opioid or insulin dosage during treatment or even performing some small tasks during surgery[23, 225].

- The system environment. Surrounding elements might affect the system's operation and safety. The kind of supervision the system will have in its environment needs to be taken into account. In hospitals, for example, we have a wider range of caregivers and surveillance than in home care.

In this sense, the task of defining safety barriers and risk classes is a complex engineering task that depends on diverse elements spread over several stakeholders. The above list is

not complete, and it is not the focus of this thesis to define a complete engineering process to complete it. However, we have provided guidelines and models that speed up the process for medical and cooperative CPS.

After defining the risk classes and their respective representation for situational risk, we need to clearly specify the values for quantitative risk evaluation at runtime. This is an empirical task because it depends on experts for analyzing and defining the risk level for each class of situation. Moreover, the quantitative aspects depend on the implementation technology used for risk assessment.

In this thesis, we employed a risk assessment approach that can be implemented using Bayesian Networks for assessing the current risk as explained in the previous chapter. At the end of the process, we applied a defuzzification algorithm in order to obtain a final number for the risk evaluation. Hence, we have tuned the best value ranges for the risk regions with experienced medical experts using several simulations and evaluations of representative usage scenarios. In Table 4, we present the risk ranges for the case study.

Risk Class	Range
Catastrophic / Critical	>= 65
Serious	45 >= && < 65
Minor	27 >= && < 45
Negligible	0 >= && < 27

Table 4 Risk classes and risk value ranges

For the case study, we defined a range for the risk assessment values from 0 to 150, where low risk values represent a lower probability of overdosage and higher values mean that an overdosage is imminent. In this sense, the catastrophic and critical range values are the widest in order to meet the real situation model where we have the most numerous occurrences of situations in these risk classes. Next, we split the classes into ranges to represent the conceptual model presented in Figure 45.

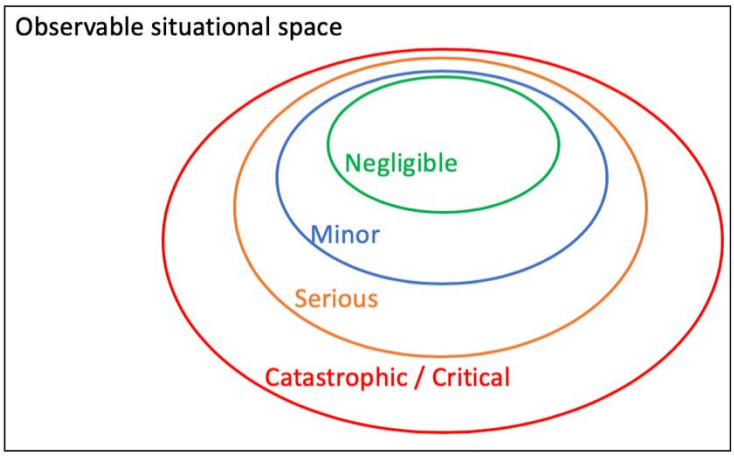


Figure 45: Conceptual split of situational space

5.3.2 Specifying the transition between risk classes

The next step is to understand and specify how situational risk behaves in the established risk classes. Like the whole risk evaluation process, it totally depends on how the risk assessment was established before. Hence, both the chosen risk scale and the risk calculation will determine the division of the risk classes and the transition between them.

Another fundamental prerequisite for determining the transition between risk classes is to understand the timing between the risk measurements. For instance, a centralized window time technique shifts the control of the risk calculation time window to a unique module responsible for managing the clock and proceeding with the risk assessment given all the sensor data, system and context states. On the other hand, it is possible to choose continuous risk assessment, where the risk calculation is triggered every time the system perceives a new change in a monitored risk-relevant element. However, we can specify how the risk value will behave and guarantee control over the risk class transitions.

We must assure that the risk class transitions will be performed in a discretized way in order to avoid that risk classes are skipped. For example, in a four-classes risk

evaluation, the risk value must not jump from risk class I directly to risk class IV. Such behavior would undermine all risk barrier plans for reducing and mitigating the ascending risk value.

In order to realize all the blueprints and requirements described, we present in Figure 46 the specified transition model for the running example. First, as we will detail in the next section, we decided to implement a centralized time window risk assessment approach with a discrete risk scale. For this example, we decided to use a risk scale between 0 and 120, due to the implementation details of the risk assessment. Then, we split the situational space using the discrete values explained in the previous section and established the transition model depicted in Figure 46.

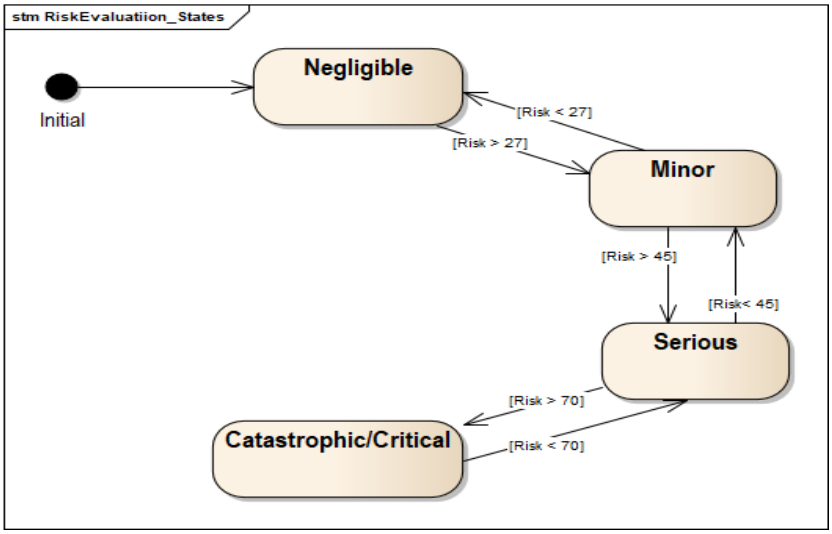


Figure 46: State transition model for situations

5.4 Risk control countermeasures

Standards of all regulated industries require a set of procedures to reduce the risk to a reasonable level. Therefore, manufacturers need to show evidence that risk analysis was performed, and which kind of measures were implemented to reduce risks to, or maintain them within, specified levels.

In this sense, such measures are organized in the literature according to their application in the process. Considering the "Bow-Tie" model, safety barriers (measures) were defined as physical or non-physical means intended to prevent, control, or mitigate undesired events or accidents [81, 191, 214].

In Table 5, we show an overview of the application of the risk barriers according to their application in the whole system lifecycle. Initially, prevention focuses on avoiding the causes that lead to a hazardous situation by strengthening the system development such as fault tolerance techniques, quality assurance approaches, specification and implementation of safety requirements related to the product, and so on. Most of the prevention techniques are related to activities performed at development time, i.e., when the system design can be modified and is under the manufacturer's control. These measures have been widely adopted by several regulated industries because standards and accreditation institutions require evidence regarding especially this kind of risk control measures for certifying products. However, these kinds of measures based on the system design can easily become outdated (or sometimes even invalid or useless) because of the challenging dynamicity of cooperative MCPS.

Furthermore, mitigation barriers are related to protecting the targets from harm once the unwanted event has already happened. Usually, they are defined as the last contention barriers for mitigating the consequences of an event and must work independent of and concurrently with prevention and control measures. Such measures can be implemented physically, such as contention walls against radiation emitted by an X-Ray machine, or defined by standards and human operators, such as clinical procedures for recovering a patient from overdosage. Since these kinds of barriers are therefore defined by a wider set of safety analyses that comprise the whole organization and are seldom controlled by the MCPSoS, we focus our definition of risk control measures on the control functions.

The control function of safety barriers prevents the transition from lack of control to loss of control [81]. This definition

perfectly matches the concept of ALARP for dynamic and autonomous tasks performed by MCPSoS. Autonomy and dynamicity challenge safety assurance at design time due to the unfeasibility of analyzing and assuring all potential system and context variations at design time. However, control function barriers are specified by the current risk once we have identified tolerable, intolerable, and unacceptable situations so that we can allocate each measure according to the correspondent risk.

Function	Definition	Example	Allocation
Prevention	Aims at suppressing all potential causes of an event by changing the design of the equipment or the type of product used	Safety requirements as defined by IEC/ISO 61508	Development time
		Pre-defined adaptation models	Development time/ Runtime
Control	Aims at limiting the transition from a normal situation to an unacceptable one	Safety Interlocks (SiS)	Development time/ Runtime
		Autonomous actions	Runtime
		Alarms	Runtime
		Dynamic adaptation	Runtime
Mitigation	Aims to protect the environment from the consequences of an unwanted event that has already occurred.	Operator actions	Beyond system boundaries
		Plant alarms	Beyond system boundaries
		Walls	Beyond system boundaries

Table 5 Safety barrier function allocation according to product lifecycle.

5.4.1 Countermeasure identification

The identification of risk measures must consider several stakeholders and system artifacts to achieve completeness with respect to the hazardous situation and the feasibility of implementation by the system.

First of all, we need to organize all the required information for the identification of risk measures. To do so, all of the presented risk assessment and evaluation needs to be specified and formalized. Based on the risk assessment and evaluation, the top-level safety requirements need to be stated and defined to avoid hazardous situations.

The next source of information for the identification of risk measures are the stakeholders. Experts need to be consulted to specify what actions the system should/could trigger given the use cases in order to avoid any unwanted event. They also need to contribute the clinical procedures, hospital protocols, and standards that rule the whole procedure in which the system will operate.

Finally, the architectural and functional system view should be considered so that we can understand the system interactions, the classes of configurations, and the boundaries. Through the analysis of the actuators, for example, we can specify how the risk control measures will be implemented.

According to the literature, the main types of safety barriers for functional control are safety interlocks, alarms, operator supervision and/or intervention [81, 191, 214] . Moreover, we have observed that the dynamicity of MCPS have enabled advanced controls such as autonomous actions and dynamic adaptation [130]. We must hence understand which risk controls we can implement and how they will be implemented in the system.

For the case study presented in Section 4.1, we considered the specified safety barriers actions, alarms, and system adaptation. Currently, there is no infusion pump on the market capable of accepting remote dosage regulation that would

allow implementing an autonomous action for reducing the risk of overdosage. Therefore, we have the following risk control actions:

1. **Alarms**
    a. **Functional alarms** – alarms need to be raised by the monitoring devices.
    b. **Risk alarms** – overall alarms need to be raised according to the overall risk of overdosage.
    c. **Adaptation alarms** – message needs to be sent to the caregivers to demand a new system configuration given the current situation.
2. **Safety interlocks**
    a. **Disable bolus** – stop the continuous infusion.
    b. **Enable bolus** – resume the continuous infusion.
    c. **Disable basal** – stop the additional infusion.
    d. **Enable basal** – resume the additional infusion.

5.4.2 Countermeasure allocation

We defined risk measure allocation as the process of establishing which risk measures will be triggered when the situational risk level rises. Considering that safety interlock measures might affect treatment availability, we must only activate safety actions when it is really necessary. For example, in our case study, disabling bolus or basal dosage will forbid opioid infusion, which will eventually increase the patient's pain level. Hence, we distributed the risk measures according to the risk classes defined above.

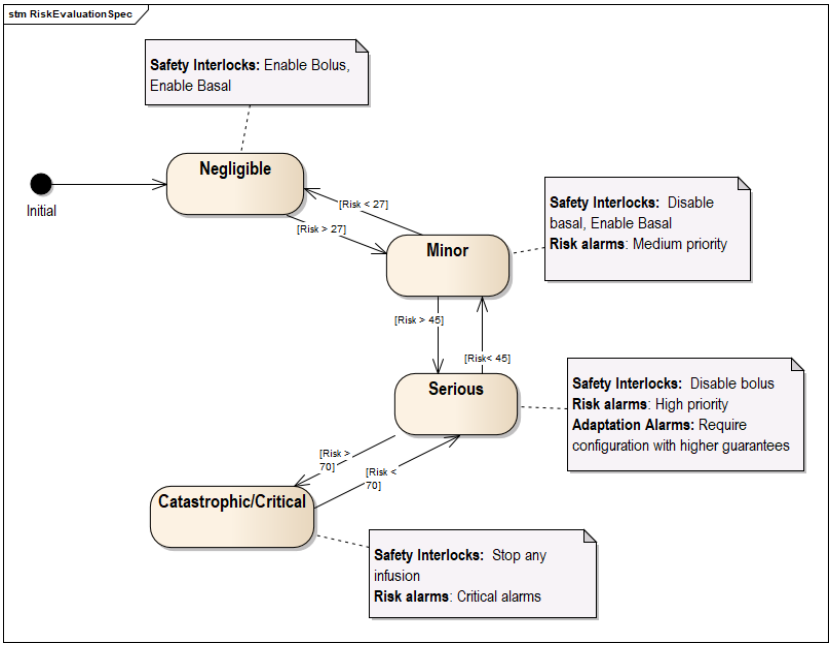


Figure 47 Risk measure allocation for MCPS

In Figure 47, we show the proposed allocation solution for the risk measures and the risk evaluation. When the situation is deemed a Negligible risk, the safety interlocks activated are Enable Bolus and Enable Basal Dosage for the patient. Hence, the patient is allowed to have all the predefined dosages delivered by the infusion pump. Then, when the risk level increases to the Minor risk class, we defined safety interlocks for disabling basal dosage and enabling bolus. Hence, the patient cannot have a continuous dosage unless the risk level decreases back to Negligible. This may happen, for example, by enhancing the system configuration or changing the context somehow to ease the risk level. Moreover, risk alarms must be raised to caregivers with medium priority to enhance the awareness level for the situation. The Serious risk class demands even more restrictive actions. Bolus dosage is not permitted, risk alarms are raised with high priority, and adaptation alarms are raised to directly require higher guarantees for the system configuration. As this risk class defines the last barriers to avoid an imminent overdosage situation, the system and the caregivers should avoid any increase in the risk level. Finally, the Catastrophic/Critical

risk classes are defined to raise critical alarms in order to have caregivers perform any mitigation actions for overdosage. Also, the infusion pump must be completely shut down as it should not deliver any doses to the patient under the system commands.

In Figure 48, we show a UML activity diagram specifying the risk management algorithm for activation of the risk countermeasures. It is worth noting the different colored elements in the picture: In pale yellow, we have all the safety interlocks; in light green, we show all the risk alarms; in light pink, we present the adaptation alarm; and the light gray elements define the risk assessment tasks.

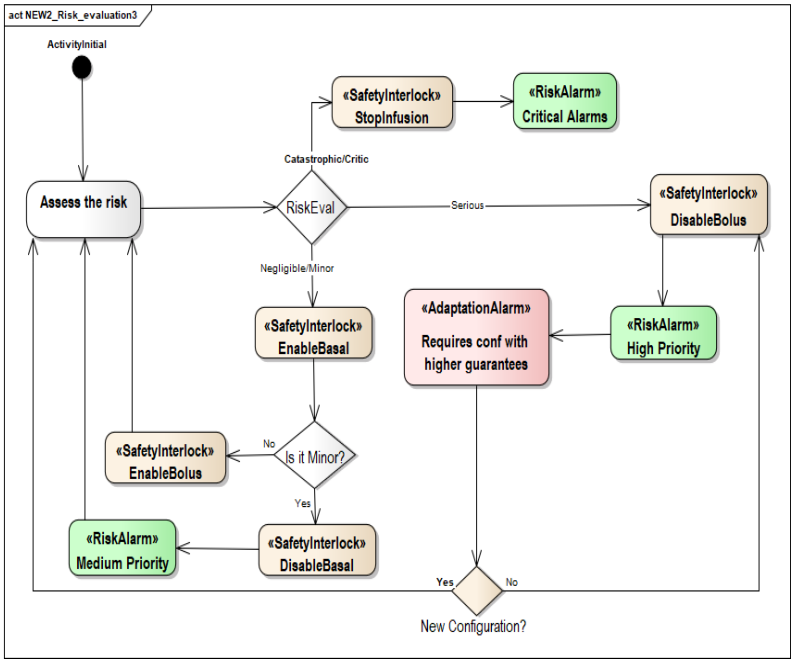


Figure 48 Algorithmic view of runtime risk evaluation and countermeasures activation

Initially, we defined a system module responsible for continuously monitoring the risk level and then performing the risk assessment tasks defined in the previous section. Thereby, if the risk is classified as Minor or Negligible, it keeps the bolus/basal enabled and sends the command to the infusion pump. Then, if the risk level is deemed Minor, a Disable Basal command must be sent to the infusion pump

and an alarm with medium priority should also be raised for the caregivers. Furthermore, if the risk level is deemed to be Catastrophic or Critical, the infusion should be stopped, and the caregivers must be warned with a critical alarm. If the risk level reaches the Serious level, a Disable Bolus command should be sent to the infusion pump and a reconfiguration alarm is sounded to the caregivers so that they can reconfigure the system to activate a configuration with a higher level of confidence. Hence, the safety monitor will only allow subsequent infusions if a new reconfiguration has been set.

5.5 Extended integrated clinical environment architecture for safety monitoring of medical cyber-physical systems

The reference architecture for the Integrated Clinical Environment (ICE) (ASTM F2761-2010) defines the role of several functional modules for implementing medical cyber-physical systems. As explained in Chapter 2, this standard establishes responsibilities and interactions between modules as well as organizations for modules at a high level of abstraction. According to the standard, the ICE is the ecosystem that performs the specified system functions. It comprises the ICE manager module and medical devices connected to the system through specific communication interfaces between devices and system. However, due to the highly abstracted description, this specification lacks several details regarding the specification and allocation of risk management tasks. In this subsection, we will further extend the reference ICE architecture in order to detail responsibilities and functions, specific modules, and interactions in order to deal with the risk management tasks defined in this thesis.

Considering the state of the practice for MCPSoS in healthcare institutions (such as hospitals, home care providers, and so on), we shall assume that a wide range of operators (caregivers, technicians, accompanying persons, and so on) can control the reconfiguration process. For instance, a nurse can add a respiration monitor to a patient during treatment or remove it, and the patients themselves can easily disconnect a pulse oximeter or other sensors.

Moreover, our approach assumes a set of configurations previously defined by the literature and clinical standards. The module responsible for performing safety monitoring must be capable of demanding a new configuration when necessary. In this way, the safety monitor must be aware of the current configuration, its capabilities, and its guarantees. In this sense, runtime certification approaches such as [168, 194, 204] have been considered as fundamental support for establishing runtime assurance of non-functional properties, e.g., safety for new system configurations. In ConSerts [203], for instance, each abstract configuration must have a ConSert Tree (CST) describing the guarantees and demands for each required system service. Besides, a distributed algorithm analyzes the safety capabilities of all cooperating systems of the cooperative MCPS in order to establish the current overall safety certificate. Therefore, we can assume that the cooperative MCPS will always have a valid safety certificate stating the system guarantees according to the current configuration because otherwise the system would not be permitted to operate.

In Figure 49, we present the extended ASTM ICE manager architecture with the respective modules and interfaces. The function of each component is described below:

- ***ICE supervisor*** – responsible for processing the risk model (presented in the previous chapter) in order to assess and manage the risk at runtime.

- ***ICE network controller*** – responsible for ensuring that the functional capabilities are in accordance with the non-functional capabilities. Hence, this component holds a set of CSTs that describe all potential configurations that the system might activate and certify at runtime. Once the configuration is selected, it triggers the runtime certification process in the configuration manager.

- ***Configuration manager*** – This component manages the current configuration, such as the runtime certification process. It is responsible for coordinating the certification algorithm and

triggering the matching of the devices' certificates to the required configuration guarantees.

- ***Device controller*** – responsible for dealing with the actuation communication interface of the devices that control the treatment, e.g., the infusion pump.

- ***Alarm system*** – responsible for managing alarms according to the requirements of the standard IEC 60601-1-8.

The runtime certification process can assure the safety of the contracts for the selected configuration. In the example of Figure 49, there are three different abstract configurations: configuration 1 Guarantee-A (lower safety guarantees); configuration 2 Guarantee-B; and configuration 3 Guarantee-C (higher safety guarantees). The hospital staff can switch between any potential configurations. The literature and practical experience show that the most common method for PCA treatment is configuration 1. Therefore, the configuration manager is responsible for monitoring any changes in the running system and notifies the ICE network controller so that it triggers the recertification process according to the respective CST. We specify the ICE supervisor component such that it can improve decision-making in terms of (1) changing the configuration to better support PCA treatment and pain relief; and (2) deciding when the opioid infusion should be provided to the patient. The risk control actions must be specified according to their integrity level and the different risk levels measured for the situation. For the safety interlock scenario, we consider three different types of actions:

a) **Sounding alarms** – *Adaptation alarms* are sounded when a new configuration is required; in addition, *Risk alarms* are sounded to warn caregivers about the situational risk level.

b) **Disabling infusion** – an intermediate risk control action to disable further required infusion that does not require sudden stopping of any ongoing infusion.

c) **Enabling infusion –** Infusion is enabled when the risk level is suitable for this command.

d) **Stop infusion** – Stopping any ongoing infusion is the action with the highest integrity level because it is associated with the most critical risk levels. We assume that taking this action implies disabling any required infusion and sounding an alarm with the highest level of criticality.

The dynamic risk evaluation approach monitors the risk at runtime and then takes control actions based on the risk levels and the current configuration. The basis for the risk evaluation is the configuration-dependent risk metric.

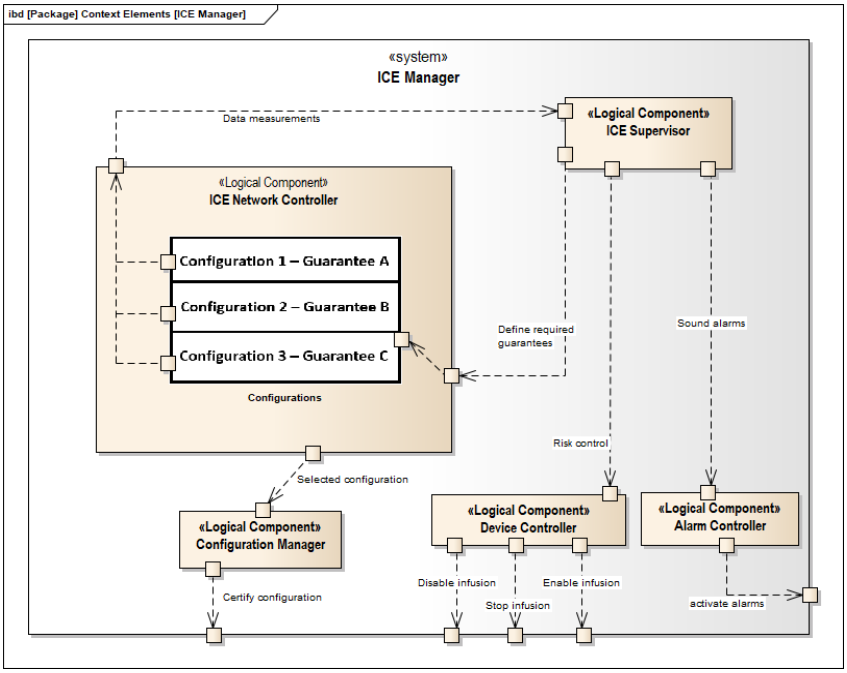


Figure 49 Extended reference architecture for ICE

5.5.1 Limitations for hazard and risk analysis completeness

Traditional HaRA techniques often assume that the complete system design and all the devices were defined at design time in order to perform safety assurance techniques. Such techniques initially define preliminary hazards through an analysis of the elements that can be a potential source of harm based on the investigation of the system environment and user interactions (PHA). Furthermore, system functions are investigated to identify and analyze potential hazards based on classes of possible failures that might occur and trigger any harmful chain of events. For instance, an infusion pump that provides an infusion function defined by a dosage and time restrictions might exhibit common failures such as late or early infusion, omission or commission, and wrong dosage values.

This assumption frequently jeopardizes the safety assurance and certification as we cannot completely analyze at design time which failures in the realization of the final system functions may cause critical failures, due to their dynamicity and adaptability at runtime. Thus, we can hardly come up with a safety concept that appropriately addresses these failures with fault avoidance, fault tolerance, and fault removal.

The first step towards the elaboration of a complete risk parameters elicitation is to define the notion of completeness of the risk parameters with respect to the identified hazards and the foreseeable usage of the system.

6 Evaluation

In this chapter, we will present an overview of the different validation and evaluation activities that we conducted during the development of the presented approach. The results were used as drivers for the continuous evolution of the proposed models and techniques. At the same time, the validation activities were refined according to the progress of the approach.

The focus of the evaluation activities was on assessing practical aspects and the performance of the proposed approach. We therefore defined two important perspectives to be considered in relation to this question:

1) Feasibility of the core concepts and the overall methodology. This comprises all the tasks regarding the proof-of-concept model.

2) Evaluation of the relevance of the approach for the state of the art. This comprises the tasks that analyzed the overall behavior of the proposed approach given different scenarios and comparison with other approaches.

Hence, we structured this chapter according to these perspectives. In the first section, we will present the proof-of-concept validation and in the second section the extensive evaluation of the simulated experiments.

6.1 Implementation and evaluation of a proof-of-concept prototype for self-adaptive risk management architecture

6.1.1 Evaluation design

The main aim of this first evaluation project was to prove the feasibility of the risk metric, the proposed risk monitoring architecture, and the risk management algorithm. Therefore, we conducted an analysis of the dynamic risk assessment model for the purpose of checking feasibility and relevance from the

perspective of safety engineers, patients, and caregivers in the context of providing care for patient-controlled analgesia treatment through cooperative medical cyber-physical systems. This evaluation is presented in the publications [leite2017, leite2018].

Hence, we considered the following research questions for driving the evaluation process:

RQ1. Can we enhance the runtime risk assessment for PCA? Fixed-threshold approaches struggle with identifying fuzzy critical scenarios that are deemed critical by the experts but have not violated the established borders yet. Our assumption is that the proposed approach will improve the identification of critical situations due to its more complete risk metrics.

RQ2. Are the identified elements relevant for dynamic risk assessment of MCPS? Considering the risk metrics defined with the support of a medical expert, we analyzed the relevance of the risk parameters, their conditional probability tables, and the risk calculation in the Bayesian network to model the risk behavior of our dynamic risk assessment approach.

RQ3. What is the impact of system reconfigurations on risk assessment for MCPS? Considering the defined risk metrics, we evaluated the impact on the risk caused by adaptations of the system. We considered the PCA architectural model (see Section 4.1), which might work in three potential abstract configurations.

For the evaluation process, we considered the scenario defined in Section 4.1, which refers to a patient in a hospital ward under a regular, post-surgery recovery PCA treatment supported by MCPS. The MCPS can assume three different abstract configurations, which vary according to the available sensors and their respective service guarantees as well as the delivery device (the infusion pump). During the treatment, the physicians might define several variations of the treatment based on the patient's status, the surgery, and the patient's history. Hence, we defined a set of complex variables to support the evaluation of the analysis and the simulation, which are described in Table 6.

Research Question	Variables		
	Independent	Confounding	Dependent
RQ1	Vital signs, context parameters, configurations		Risk
RQ2	Vital signs, context parameters, configurations		Risk, sensitivity measure
RQ3	Configurations	Vital signs, context parameters	Risk

Table 6 Variables analyzed for the evaluation according to the respective research question

In Table 6, we correlate the research questions and their respective evaluation variables, which can be independent, cofounding, or dependent. It is worth observing that these are complex variables composed of several atomic ones; for example, vital signs stand for the whole set of parameters that monitor the patient during PCA treatment, such as blood saturation, exhaled carbon dioxide, heart and respiration rate. Likewise, configurations define the current system configuration running the MCPS, while context parameters define treatment aspects such as patient history, additional treatments, drug type, and so on. For the evaluation of research questions 1 and 2, we controlled the variables of vital signs, context parameters, and configurations in order to observe the behavior of the final risk measured by the model. Particularly, for research question 2, we also observed the sensitivity parameter measured by the tool where we implemented the Bayesian network risk metric. For research question 3, we measured the risk behavior when simulating the system reconfigurations in order to evaluate the impact of the adaptations on the risk; in this case, the patient's vital signs and context parameters were considered as independent elements in the simulation.

6.1.2 Evaluation and Results

To evaluate the proposed approach, we performed simulations in two different environments under the supervision of a medical expert. In the first subsection, we will present the BBN evaluations done by means of sensitivity analysis and comparative analysis of the simulation results considering a concrete scenario. In the following subsection, we will present a simulation environment integrated with the OpenICE framework to simulate MCPS scenarios and patient profiles.

Bayesian network design and analysis

To design and evaluate the Bayesian network models, we worked with the academic license of the GeNIe Modeler. The tool provides several features such as a graphic tool for BBN development, analysis, function processing, and so on. Moreover, it provides the SMILE Engine for designing and reasoning through programming languages such C, Java, Python, etc. In particular, we utilized this engine to handle the nodes at runtime and to reason about the risk in the risk monitor. In Figure 50, we show a version of the Bayesian risk model in the GeNIe Modeler tool. It can be seen that we used the tool for tuning the nodes and their respective NPT in order to do the best calculation of the risk representation model.

In Figure 51, we present the specific environment for sensitiveness analysis in the GeNIe Modeler tool. The built-in sensitivity analysis function of GeNIe varies each node over the whole range and assesses the impact of this change on the target node. In this case, the target node for the sensitivity analysis is the Overall Risk Function node. In Figure 51, we show the result of the analysis for the most relevant nodes that contribute to the current risk values in the node Overall Risk Function. As you can see intensive red the nodes System, System Actuation Parameters, and Actuation Treatment Aspects indicate nodes with higher influence.

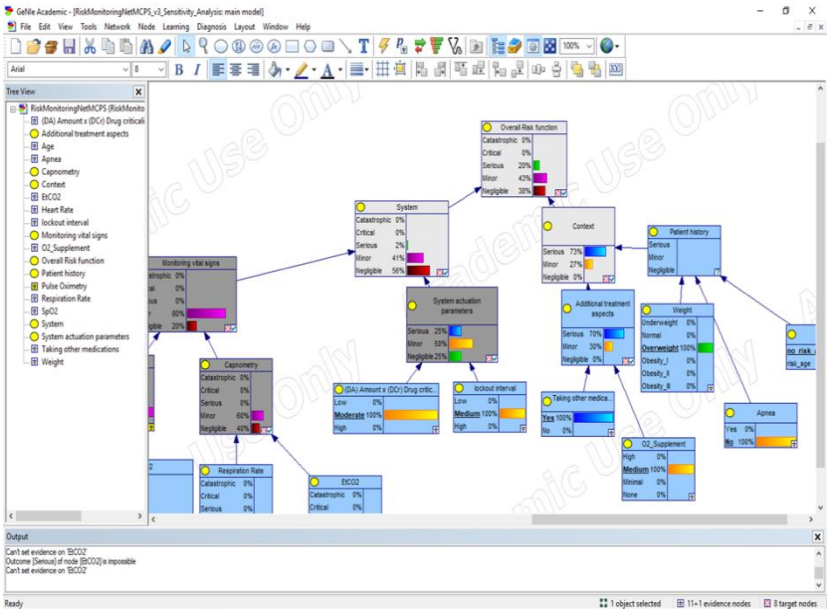


Figure 50 GeNIe modeler tool

In the figure, we can see nodes colored in pink (Context and Patient History), which are still relevant for the overall risk function. At the left bottom of Figure 51, we can see the tornado diagram showing the list of the most representative nodes and their respective values that are determinant for the risk value. We can notice the relevance of the Dynamic Data elements, which are variations of the System Data node values.

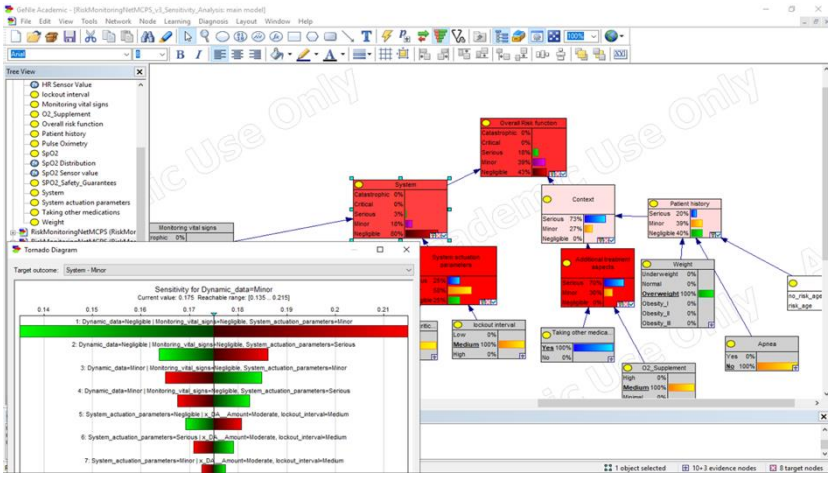


Figure 51 GeNIe modeler performing sensitiveness analysis

Furthermore, we analyzed the strength of influence based on Figure 52. The arcs have different thickness, dependent on the strength of influence between the nodes that they connect. Strength of influence is always calculated from the CPT of the child node and essentially expresses some form of distance between the various conditional probability distributions over the child node conditional on the states of the parent node. In Figure 52, we can observe that the most influent element for the root node (Overall Risk Function) is the System Node. The color of the nodes is not relevant for the strength analysis. The System Node is then influenced most by the Monitoring Vital Signs node, as previously defined. For the Context node, the most influential element is the Additional Treatment Node.

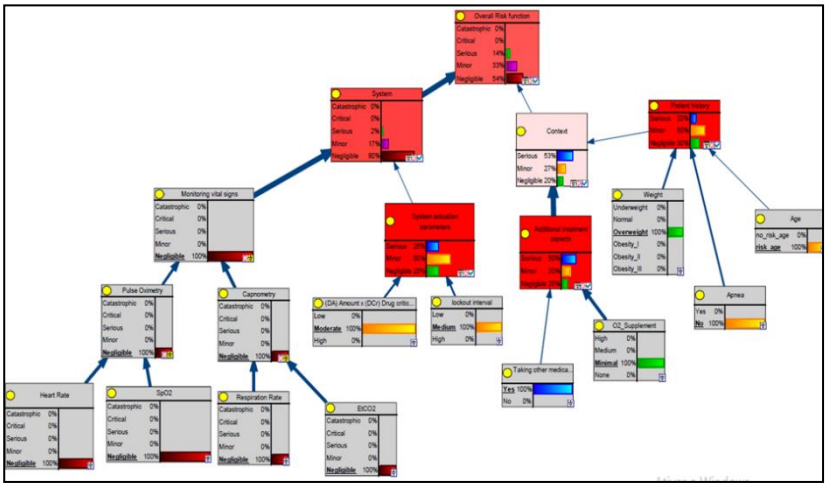


Figure 52: Strength analysis of the nodes.

Simulation environment

To support the validation of the approach with the medical staff, we developed a Java simulation environment to evaluate the results from the ICE safety supervisor and the risk assessment model. We intended to evaluate the proposed approach in a realistic simulated environment considering different patient profiles and configuration modes. The architecture of the simulation environment is shown in Figure 53.

For the simulation of the MCPS component systems and the patient variables, we worked with the OpenICE[1], which defines an integration architecture for healthcare IT ecosystems through a distributed system platform for connecting network nodes with each other. OpenICE deals with several technical issues such as node discovery, external interface definition, data publishing, proprietary protocol translation, and so on. Therefore, there is communication between the OpenICE component and the risk monitor that updates all the relevant information about the medical devices and systems, which is represented in the diagram by the SimulOutputPort and the simulation data intemFlow to the DataReadePort. Furthermore, we also needed extra support for the simulations due to the fact that the OpenICE does not work with the whole set of parameters that we must check for the risk metric calculation. Hence, we defined the Auxiliar Simulator for handling context parameters as well as the guarantees of the connected medical devices.

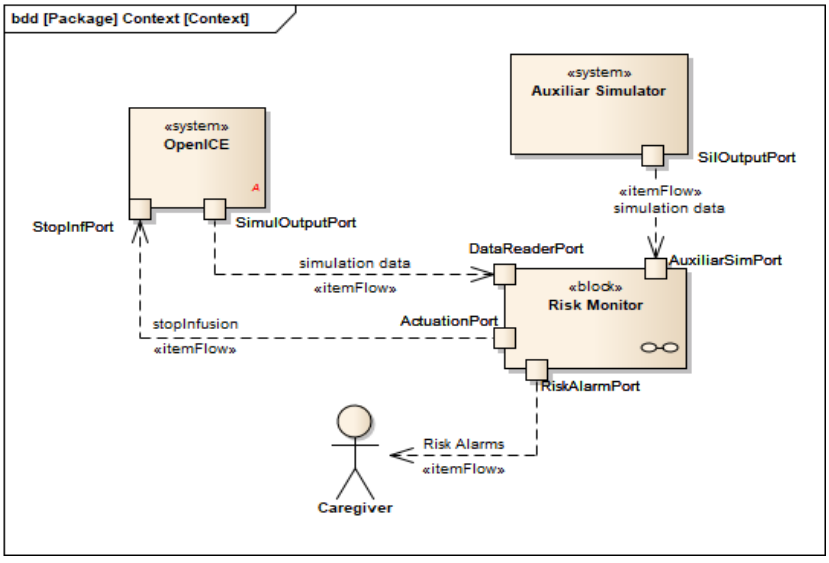


Figure 53 Architecture of the simulation environment.

Finally, we implemented the Risk Monitor to run the presented algorithm for dynamic risk assessment as well as the risk assessment model for MCPS. This component is composed of a graphical interface for risk representation and notification and

[1] https://www.openice.info/

also triggers the stopInfusion commands for the OpenICE component through the ActuationPort and the stopInfusion command.

Figure 54 presents an OpenICE GUI for the safety interlock application, which simulates the infusion and defines some alarms and stop infusion interlocks based on respiration rate, heart rate, and blood saturation (SpO2). In this way, the infusion interlocks are based on fixed thresholds (horizontal configurable sliders in the figure) that we defined together with medical experts. This baseline approach implements a very simple risk metric with a higher level of integrity due to its simplicity and precision in detecting critical situations; however, it does not realize when a situation is heading to a critical status and might thus fail to evaluate such situations (emerging risk) and also constrain the treatment when the patient could actually get an infusion with a more confident configuration. Moreover, the baseline approach does not consider the provided integrity level of the services nor the confidence level of the detection method, meaning that pulse oximetry and capnometry are considered with the same level of confidence. Therefore, there is no reasoning in the risk metric about eventual reconfiguration, which frequently occurs in this kind of treatment.

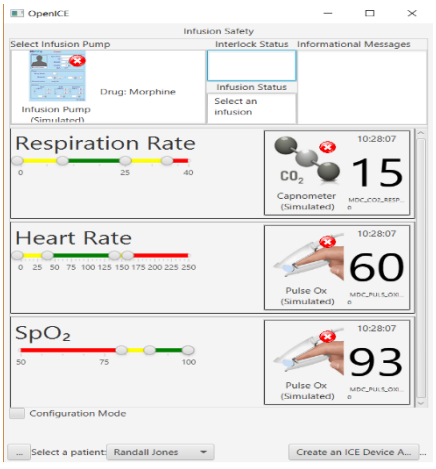


Figure 54: OpenICE graphical interface of the infusion safety application

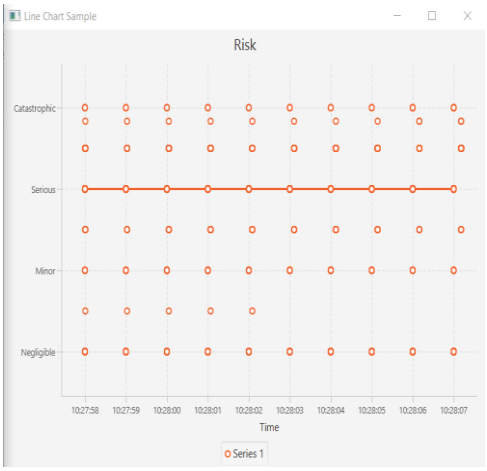


Figure 55: Graphical User Interface of the ICE supervisor indicating a Serious risk level.

In Figure 55 we show the first version of the risk monitoring interface. The caregivers can check the current risk classification watching the timed risk curve represented in the line chart. In. The Figure 55 shows the defuzzied risk value classified according to the five classes of risk and the time. We can observe that the risk level is classified as Serious, whereas there is no indication of any alarm or measure in Figure 54 (baseline comparison). The risk metric has a lower integrity level due to the complexity of the risk assessment implementation; however, we improved the identification of risky situations (emerging risk). This fact is due to our risk model considering a wider set of parameters, their relationships, and their weights. Moreover, our risk assessment model considers dynamic adaptation and guarantees provided by the active configuration; thus the risk metric varies according to the reconfiguration, evaluating the current risk and sounding alarms to the caregivers as specified in Chapter 4.

In order to use a controlled scenario, we simulated a patient in post-operative PCA treatment and evaluated various respiration rate values measured by the capnometer. We validated the results in workshops with the Anesthesia, Intensive Care, Emergency Medicine, and Pain Medicine departments of the Westpfalz-Klinikum hospital in Kaiserslautern, Germany.

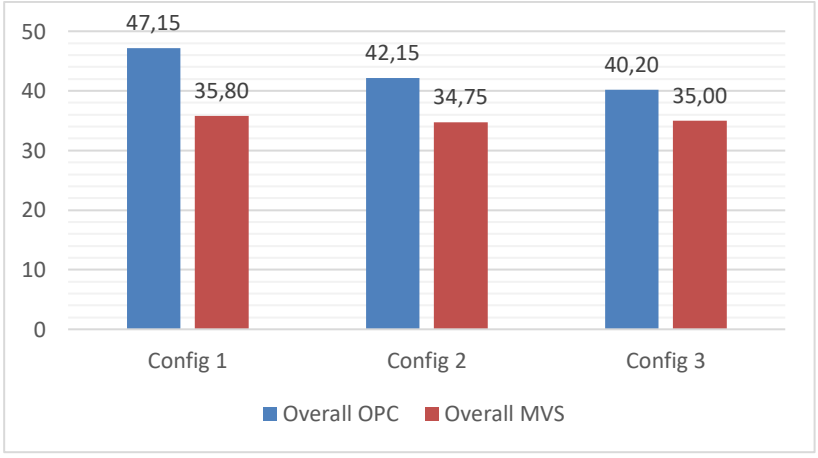


Figure 56 Comparison of defuzzied values of the Overall Function Node (OPC) and the Monitored Vital Signs (MVS) nodes of different risk metrics concerning the corresponding configurations

We also evaluated the impact of different configurations on the risk assessment. The main aim was to show that risk metrics based

on less confident methods imply a more conservative risk assessment and to demonstrate how the integrity level affects the risk evaluation. In that sense, we evaluated a similar scenario for the three different risk metrics regarding the presented configurations, which were based on a controlled patient (stable pain level) with medium oxygen supplementation and the vital signs monitored having Negligible and Minor status. The values presented in Figure 56 were defuzzied using the Weighted Average Method [189] for the OPC and MVS nodes. Then we classified the risk level according to our five risk classes. For configuration 1, the current risk measure was classified as Serious due to the defuzzied risk value reaching 47.15, which, according to the domain experts' expectations, was due to the oxygen supplementation. For the same situation, risk metric 2 reached 42.15 (Minor risk value classification) for OPC, which is lower than the value measured by the configuration 1 risk metric and also conformed to the domain experts' expectations. Finally, for configuration 3, the measured risk value was 40.2, which is the lowest risk level due to the higher integrity level of the measurement, which combined capnometer and pulse oximeter readings. It is worth noting the similar values for the MVS node in the evaluated scenarios; however, these values show the relevance of other contextual information (such as additional treatment nodes and patient history). We can observe the difference between the OPC and MVS in each scenario: In the less confident risk metric (configuration 1), we gave more relevance to the contextual information, whereas in configuration 3, we considered contextual information less relevant.

6.1.3 Discussion

The first version of the BBN dynamic risk assessment was developed to validate the assumptions and requirements collected from the engineers and the safety and medical experts. Therefore, we will revisit the research questions in order to evaluate the system behavior accoding to the analyzed research questions.

RQ1. Can we enhance the runtime risk assessment for PCA? According to the results described in the section 6.1.2, the current state of the practice considers fixed thresholds and might thus miss risky situations close to the acceptable limits, which can

quickly evolve into an unacceptable risk situation. Moreover, when we consider a wider range of relevant parameters, we in fact enhance the runtime risk assessment regarding either the precision of the situation assessment or the speed (or the gradient) of the risk curve given the dynamicity of the scenarios.

RQ2. Are the identified elements relevant for dynamic risk assessment of MCPS? In the sensitivity analysis, we showed the relevance of the main nodes for the risk function. We showed that the most representative elements are the Monitoring Vital Signs node and the System Actuation node; however, the Context node has a significant impact on the risk for the analyzed scenario. The Context node aggregates data from the treatment and patient history and actually enhances the precision of the risk assessment as well as the gradient of the risk behavior. In the section 6.1.2, we compared the Overall Risk Function node with the Monitoring Vital Signs nodes. In Figure 56, we observed that the lower the guarantees of the active abstract configuration, the higher the relevance of the Context node for enhancing the risk assessment. This behavior was evaluated positively by the medical experts, who confirmed it as confident support for risk assessment, especially due to the fact that the most common abstract configurations in hospital wards worldwide are the configuration that use only a pulse oximeter sensor. Hence, during the validation workshop, the safety experts and the medical experts confirmed the relevance of the risk parameters.

RQ3. What is the impact of the system reconfigurations on the risk assessment for MCPS? In the section 6.1.2, we analyzed the impact of the reconfigurations on the risk assessment. Overall, we presented the relevant impact of the different system guarantees on the risk assessment. Hence, when the active configuration exhibits higher guarantees, such as the double sensor in the abstract configuration 3, the risk metric relies more on the Monitoring Vital Signs nodes than in the other configurations. Therefore, we demonstrated the relevance of the system reconfigurations in the risk assessment.

6.2 Simulation-based experimental study

The main aim of the experimental study was to analyze the proposed DRA model for runtime risk assessment for the purpose of validation with respect to efficacy, efficiency, treatment availability, and robustness regarding system adaptations from the perspective of safety engineers and health professionals in the context of providing care for patients via cooperative MCPSoS.

In this sense, we defined the following research questions in order to guide the evaluation process for the comparative assessment of the results provided by the vital signs aggregation approaches based on state of the practice (SotP) and the state of the art (SotA):

1) **RQ1**: Can the proposed approach improve the **efficacy** of risk assessment compared to SotA and SotP? In the context of this work, efficacy is defined by the accuracy with which the risk assessment is performed for a given situation by the dynamic risk analysis approach.

2) **RQ2**: Can the proposed approach improve the **efficiency** of risk assessment compared to SotA and SotP? In the given context, efficiency is related to the time elapsed until the safety monitor detects an imminent hazardous situation through the proposed dynamic risk analysis as well as the time that the monitor spends on making precise decisions.

3) **RQ3**: Is the approach capable of enhancing **treatment availability?** In the context of this research, we must analyze whether the risk classification approach enhances the availability of the treatment for patients under PCA. Improving the availability of PCA treatment will enhance the patient's experience and decrease the average pain level during the treatment. Therefore, we identified scenarios where current approaches conservatively forbid infusion and the proposed approach allows it.

For this evaluation, we designed an experimental study comprised of interviews with medical experts, laboratory simulations, and further validation of the results. In the next section, we will present the design of the experiment in detail.

6.2.1 Evaluation design

According to [67], this study included fundamental elements of Structure Validation due to the inherent nature of evaluating a specific software element (i.e., the risk metric) and the multidisciplinary team of researchers involved in the evaluation. Safety and system engineers evaluated the suitability of the models and the feasibility of the prototype in the simulation environment, whereas medical experts evaluated the outcomes of the overall system behavior, the validity of the calculations, improvements, and drawbacks.

The experiment used for the evaluation of the overall system behavior was performed by means of simulated PCA scenarios. Hence, the experiment comprised three steps: (1) scenario definition, (2) simulation, and (3) scenario validation.

The PCA scenarios were initially derived by the medical experts based on their professional experience and on the literature. We selected six medical doctors from Germany and Brazil working in different fields: anesthesia, thoracic surgery, respiration, intensive care, knee surgery, and hip surgery. Unstructured interviews were performed to define the scenarios. We defined the following scenarios for PCA for post-operative pain relief treatment: cardiac surgery, pulmonary surgery, hip replacement surgery, and knee surgery. Each scenario was classified according to the treatment's characteristics (type and amount of opioids, necessity of oxygen supplementation, etc.), the patient's profile (age, weight, any respiratory disease), and the evolution according to the procedure. For thoracic surgery (cardiac and pulmonary surgery), more critical parameters were defined for the patients and the treatment due to the most frequent type of patients and the natural aspects of the surgery, such as pain level and recovery process. For knee and hip surgery, a less critical profile was defined for the post-operative evolution; however, the patient profiles were also diverse as knee surgeries are also common for otherwise healthy people and for athletes. On the other hand, hip replacement surgery is more common for seniors and for obese patients.

Once all the scenarios had been specified, we implemented all simulations according to the specified treatments and approaches. This environment comprised two systems. The first was an MCPS simulation implemented with OpenICE [179], which defines an integration architecture for healthcare IT ecosystems through a distributed system platform for connecting network nodes with each other. It deals with several technical issues such as node discovery, external interface definition, data publishing, proprietary protocol translation, and so on. The second system in the simulation environment was an ICE monitor, which was responsible for implementing the risk assessment for the MCPS regarding all four analyzed approaches: (1) the proposed dynamic risk assessment model; (2) a risk model for the state of the practice; (3) an aggregative risk model for the basic vital signs; and (4) a risk model for the state of the art with reconfiguration.

The scenario validation phase comprised the development of an evaluation dataset and an analysis procedure over the results. First, we had to validate the simulations with the medical experts by doing interviews with them. We presented to them the patients' responses to the treatment for every scenario with the referred outcomes such as situation identification, risk assessment, and risk countermeasures taken. Following the validation of the simulations, we developed a dataset with all measures identified for the results analysis. For this purpose, we ran the simulations and collected data on accuracy (false and true positives, false and true negatives), efficiency (respective time omission, commission, and precise measurements), availability (time to recovery, time between failures, total uptime), and robustness (number of detected reconfigurations, time between reconfigurations). Finally, we concluded the validation phase and the collection of the data and analyzed them with appropriate descriptive statistics to establish an understanding of all the data collected.

6.2.2 Results

In our work, we comparatively evaluated our dynamic risk assessment approach with techniques from the SotP and the SotA in order to determine whether our approach improves efficacy, efficiency, and treatment availability. In this sense, we would

then be able to pave the way towards automated actuations (safety interlocks) for such challenging dynamic cooperative MCPS.

For the experiment, we considered four evaluation scenarios to run simulations and gather data for analyzing the system behavior with respect to the predefined measures. We split the scenarios into discrete time windows named situations. The number of situations could vary according to the scenarios. We used these time marks for measuring the systems' performance parameters.

We considered approaches from a survey about the techniques used in industry and found in the literature. The main aim of the simulation was to prove the concept and evaluate a comparative study about the techniques, considering real evaluation scenarios identified and validated by the medical experts. In the following, we present these approaches in detail, assuming all the abstract architectures for PCA presented in Chapter 4:

1) **SotP - a risk model for the state of the practice**: We surveyed leading-edge efficient hospital solutions for patient monitoring, such as Siemens, Philips, Dräger, GE, and so on. They combine advanced and useful technologies such as hospital information system integration, integrated monitoring (for emergency control rooms), IoT applications, Big Data analysis, and so on. However, few applications enable remote system actuations such as safety interlocks (closed loop control) as assumed for the ASTM F2761 standard and considered in this work. According to the literature [146, 147] and the experience of the medical experts, the most frequently used PCA monitoring approach in hospitals worldwide is the abstract configuration 1 with pulse oximetry.

2) **SotA – an aggregative risk model for the main vital signs**: We explored the literature to identify the most commonly used abstract architectures for PCA loop control. Several solutions have been proposed, such as [155, 173], and although several works provide remote system actuation (automated safety interlocks), system adaptation is still a challenge for these types of systems. Overall works have presented valuable and promising

advances regarding accuracy, for example, control theory [13], machine learning techniques [138], Bayesian networks inference [84], and so on. Therefore, a fundamental limitation of such solutions is the fact that they assume a limited set of risk parameters and that the risk calculation method needs to consider the risk definition from the standards. We implemented a general method to aggregate in a Bayesian network the main vital signs measured during PCA treatment, i.e., blood saturation, heart rate, respiration rate, and exhaled carbon dioxide.

3) **SotA 2 – an aggregative risk model for the main vital signs (sensitive to reconfiguration):** Considering all the promising solutions from the literature review, we simulated a novel SotA application capable of detecting system reconfiguration. Even though there is no such reference in the literature, we decided to implement such a solution in order to enrich the comparative evaluation. We therefore implemented an aggregated Bayesian network for all the vital signs from the SotA implementation. This system is smart enough to adapt to reconfigurations. In this sense, if the system starts the monitoring only with a pulse oximeter, and afterwards a respiration monitor is integrated into the overall system, the data measured by this device will also be integrated into the risk metric.

4) **DRA – the proposed approach for dynamic risk assessment**: Our adaptive risk metric approach, which is capable of detecting any system reconfiguration and instantiating a proper risk metric, as presented in the previous sections. We validated our approach as presented in different publications [130–132].

In the following, we will present a descriptive analysis of the results according to the validated simulations and the statistical data analysis.

Descriptive analysis for the results regarding RQ1 – Accuracy

Research question 1 asks whether the proposed approach enhances the accuracy of cooperative MCPS. We defined accuracy in terms of the efficacy of assessing the current situational status given all the parameters monitored by the system. Hence, we analyzed the proposed DRA model and SotA/SotP risk assessment techniques for the purpose of evaluating their efficacy on situational risk assessment at runtime for patient-controlled analgesia treatment in the context of cooperative medical cyber-physical system scenarios.

For this evaluation, we defined a set of metrics in order to formalize accuracy in the context of the experiment. First, we defined the accuracy rate based on the classical definition for information retrieval [150]. It is thus given as follows:

$$Accuracy(ACC) = (\sum TP + \sum TN) / \sum (TP + TN + FP + FN)$$

Equation 5 Adapted accuracy equation for precision of alarms

According the Equation 5, accuracy is the rate of the sum of correct situation detections (true positives and negatives) over the total number of situations assessed by an approach. Hence, we collected and arranged the measures according to the respective simulation scenarios in order to calculate the accuracy rate according to scenario and approach. We present the accuracy data in Figure 57 (by simulation scenario in the columns and by approach in the rows) as well as the median, the mean, and the standard deviation for all four presented approaches.

In the Figure 57 we present the collected accuracy data (true and false negatives as well as true and false positives) for all the approaches compared through the simulated scenarios. This figure depicts our definitions of the values in terms of percentage for the referred values. We colored the true positive and true negative results blue, respectively light blue, and the false negative and false positive results light red, respectively red. We can observe that, for all scenarios, the DRA approach reached rates only for true positives and true negatives for all scenarios. For the SotP approach, we measured fewer occurrences of false positives, although it showed a higher rate for the Cardiac Surgery

scenario. The SotA approach showed a rate of 66% of true positives (better than the mean) for the Lung Surgery scenario; however, for the other scenarios, the approach struggled with high rates of false negatives and false negatives, which were even higher than the true positives and true negatives. For the SotA2 approach, we got better results for the Lung and Knee Surgery scenarios, where true positives and true negatives were higher than 70%, However, it yielded 44% false positives for the Cardiac Surgery scenario.

Figure 57 Accuracy data gathered in the simulation scenarios. True Positives (TP), True Negatives (TN), False Negatives (FN), and False Positives (FP) for: (a) Lung Surgery, (b) Cardiac Surgery, (c) Hip Replacement Surgery, and (d) Knee Surgery.

In Table 7, the differences in the accuracy results can be seen for each approach. The worst results for accuracy were registered for SpO2 monitoring (SotP), as can be seen in the mean and median values. One of the best performances among the compared approaches was definitely observed for the SotA2 (reconfiguration-sensitive) approach, which exhibited relevant positive numbers higher than 0.7 for almost all scenarios and significant differences in the median compared with the other approaches. However, the proposed DRA approach achieved the most notable accuracy for the evaluated scenarios.

	Lung	Cardiac	Hip	Knee	Median	Mean
SpO2 SotP	0,56	0,44	0,43	0,80	0,50	0,56
SotA2	0,78	0,56	0,71	0,70	0,71	0,69
DRA	1,00	1,00	1,00	1,00	1,00	1,00
SotA	0,67	0,67	0,43	0,30	0,55	0,52

Table 7 Accuracy measurements for the simulated experiments.

We can also observe a relevant variance in the measured accuracy for the simulated scenarios. The results provided for the Cardiac and Hip Replacement Surgery scenarios were remarkably limited, as can be seen in Figure 58(a), which present the graphic results for accuracy for each scenario. The figure also shows that the scenario for Lung Surgery got the highest accuracy rate mean compared with the other scenarios.

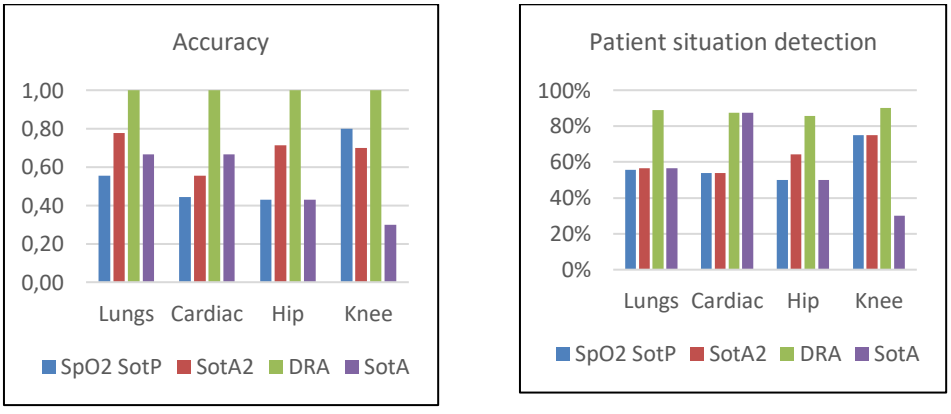


Figure 58 (a) Measured accuracy measured; (b) patient situation detection rate (PatSitDect).

The next metric was defined as a numeric value called Patient Situation Detection rate (PatSitDect), used for assessing the precision of an approach for each situation. We asked a medical expert to identify the correct assessment for every single situation in the considered simulated scenarios and distributed the score number according to the success rate of the approaches in identifying them. Hence, we defined a value of 10 for total precision of the assessment and scale values of 5, 1, and 0 according to the results of the other assessments.

Figure 58(b) presents the results for patient situation detection measured by the simulation scenarios. For the sake of a clear presentation, we transformed the scores into percentage values. Thus, it can be seen that the proposed DRA approach was successful in the simulated experiments and presented the highest values for the refereed metric. However, the compared approaches showed rather good results for Knee Surgery despite the low performance of the SotA. Overall, the average performance for PatSitDect was 59% for SotP, 62% for SotA2, 56% for SotA, and 88% for DRA.

Descriptive analysis for the results regarding RQ2 – Efficiency

Research question RQ2 is about analyzing the efficiency of the considered approaches regarding the detection of hazardous situations. We established three metrics to assess this research question in order to quantitatively compare the performance of the approaches.

The first metric targets the time elapsed until a hazardous situation is detected, the so-called PreTimeHSSitDect. We focused on the proposed DRA approach and its capacity of detecting hazardous situations, such as overdosage during PCA treatment. We defined a discrete time counting for the simulation and time metrics considering such time elapsing. Hence, we simulated some hazardous situations for all four scenarios and registered the time elapsed until the DRA approach detected the hazardous situation by raising its alarms for the Serious situation state and observed the alarms raised by the compared approaches. In this context, we recorded the logfiles of all four approaches in

order to precisely analyze the exact moment when they detected an imminent overdosage.

In Figure 59, we present the results of PreTimeHSSitDect for all the evaluated scenarios. We observed that only the DRA and SotA2 approaches had a significant advance detection time (in seconds) compared with the other approaches. The DRA approach exhibited the highest elapsed time count (82 seconds overall) for the advance detection of hazardous situations. Specifically, for the Cardiac and Lung Surgery scenarios, the DRA approach detected hazardous situations faster than all the other approaches. In total, this advance detection time was 28 seconds for the Lung Surgery scenario and 42 seconds for the Cardiac Surgery scenario (for four different simulated hazardous situations for Cardiac Surgery and one hazardous situation for Lung Surgery). Meanwhile, for Knee Surgery, SotA2 and DRA had the same performance for detecting hazardous situations in advance, with 12 seconds of prediction time for each approach.

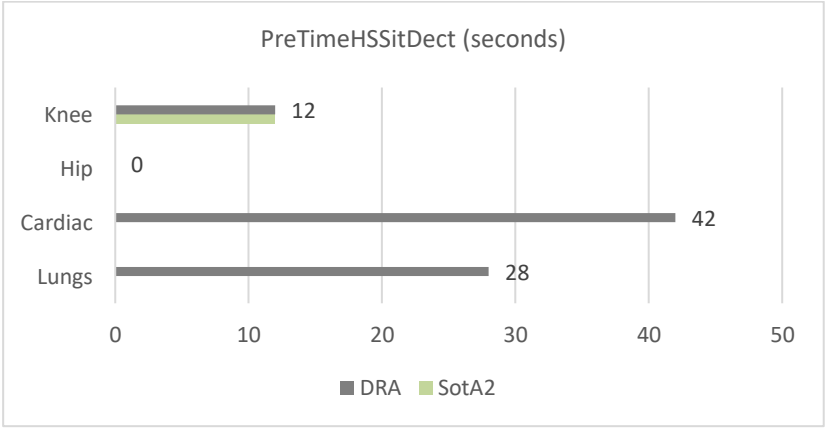


Figure 59 PreTimeHSSitDect in seconds for each scenario.

We also measured the elapsed time for each approach while it classified the situations according to the risk. The precise time metric comprises the summation of the time spent by the approach when it got a true positive or true negative. Imprecise time stands for an approach that assessed a situation with a false positive or negative.

In Figure 60, we present the results for imprecise and precise time, respectively. Observe that for the less critical surgeries such as knee and hip replacement, the SotA approach got the highest rate of imprecise time with 48% and 47%, respectively, while the same approach had 14% and 17% of imprecise time for the more critical surgeries (lung and cardiac). The SotA2 approach did not have any significant difference for all the scenarios, ranging from 13% of imprecision for Lung Surgery to 25% for Knee Surgery. Finally, the SotP approach had a smooth variation for the most complex surgeries, namely 16% for Lung Surgery and 17% for Cardiac Surgery; however, it had a lower imprecise time rate for Knee Surgery (9%) and a peak of 33% for Hip Replacement Surgery.

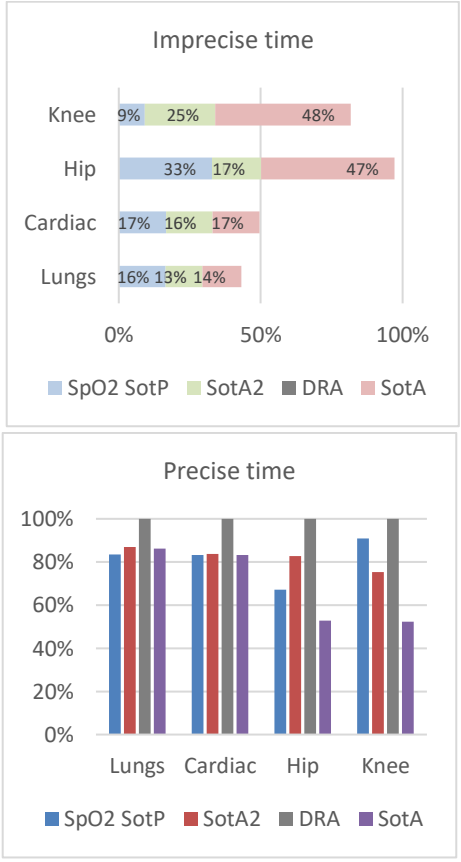


Figure 60 (a) Imprecise time rate measured by the approaches during the classification of the situations; (b) Precise time rate measured by the approaches during the classification of the situations.

The precise time measure is presented in Figure 60(b) for all approaches and scenarios. The DRA approach actually got 100% precision for all scenarios, as already observed in the accuracy data. For the most critical surgeries, all the other approaches reached a mean of 84% of precise time. However, for Hip Replacement Surgery, SotA2 still got a rate higher than 80%, while SotA reached the lowest rate of precise time with around 53% and SotP had 64%. Similarly, for the Knee Surgery scenario, the SotA approach reached 54% of precise time while SotA2 got almost 80% and SotP 91%.

Descriptive analysis for the results regarding RQ3 – Treatment Availability

With research question 3, we investigated whether the proposed approach enhances treatment availability compared with other approaches. Besides the overdosage problem, underdosage is also a problem for PCA because it may increase the patient's pain level and end up causing discomfort. Hence, treatment availability must be guaranteed during the process while overdosage must be avoided at the same time.

We defined treatment availability based on the concept of total uptime (TPU), mean time between failures (MTBF), and meantime to recover from failures (MTTR). TPU was defined as the overall time spent providing the correct treatment (opioid availability) for the patient; in other words, it is the true positive elapsed time. MTBF was calculated based on the time interval between service failures, which is considered an interruption of the treatment regardless of whether this was caused by a true negative or a false negative event. In the following formula, we considered the total time that the system provided the service correctly (TPU) divided by the number of true negative (TN) and true positive (TP) events.

$$MTBF = \frac{TPU}{TP + TN}$$

Equation 6 Definition of mean time between failure for the experiment

MTTR measures the time spent by the system during failure events (true and false negative). In the formula, TNT stands for the time spent during a true negative event and OT refers to

omission time (false negative). We calculated the rate through division by the number of true negative events (TN) plus false negative events (FN):

$$MTTF = \frac{TNT + OT}{TN + FN}$$

Equation 7 Definition of mean time to failure for the experiment

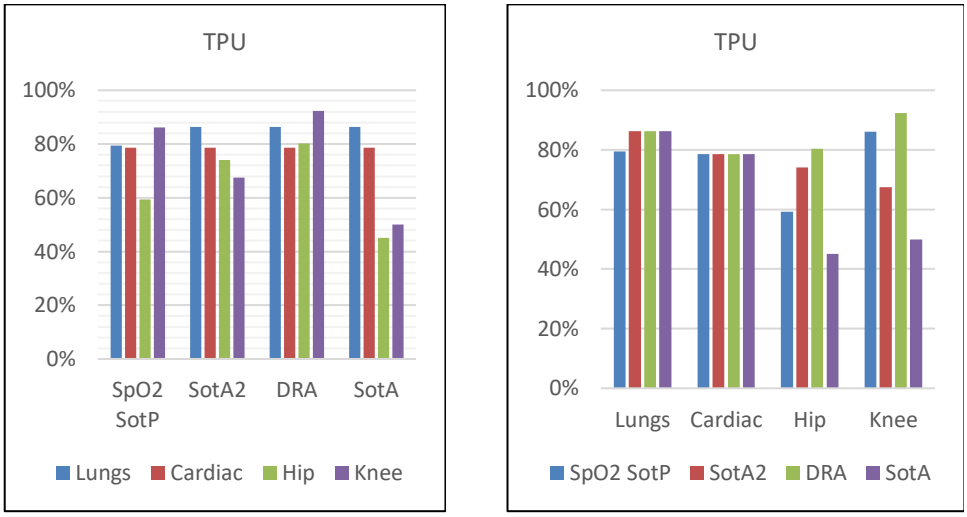


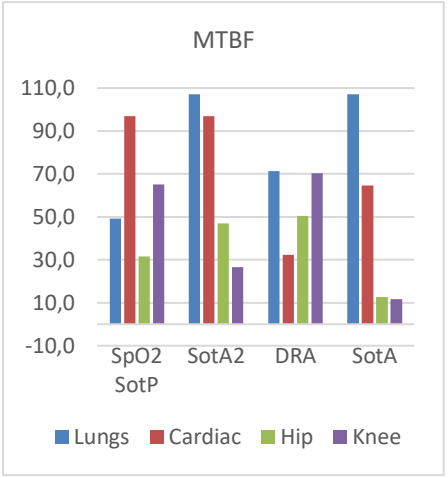
Figure 61 Total time for true positive service provision: (a) TPU by approach, (b) TPU by scenario

In Figure 61, we present the two different views for the total uptime results. In Figure 61(a), it can be observed that the overall TPU performance of the DRA approach was slightly superior compared with the other approaches; conversely, SotA got the lowest score. We verified this claim by calculating the overall mean TPU, which for SotP was 76%, for SotA2 76%, for DRA 85%, and for SotA 65%.

In Figure 61(b), it can be seen how the scenarios affect the system performance for TPU. The most critical scenarios (Lung Surgery and Cardiac Surgery) reached the highest TPU values for the approaches; for example, Cardiac Surgery got 79% for all approaches. On the other hand, the less critical surgeries provided a higher variation level for TPU; for instance, the standard deviation for Hip Replacement Surgery was 14% and for Knee Surgery 17%, while for SotP it was only 3%.

In Figure 62, we present the results for mean time between failures and mean time to recover from a failure. Figure 62(a) shows that the SotA2 and SotA approaches provided the longest mean period between failures for the Lung Surgery scenario (around 107 seconds). Likewise, the SotA2 and SotP approaches exhibited the highest mean between failures for the Cardiac Surgery scenario, around 97 seconds. The DRA approach achieved the best results for the Hip Replacement Surgery and Knee Surgery scenarios, although it had lower performance for the Lung Surgery and Cardiac Surgery scenarios. Overall, it is also worth noting that the performance of all approaches for the most critical scenarios outperformed the MTBF for the less critical scenarios.

In Figure 62(b) we show the accumulated results for mean time to recover from a failure for every scenario. It can be seen that the DRA and SotA approaches reached the highest accumulated values for MTTR with a slightly positive difference to the SotA approach. While DRA performed with a MTTR mean of 12 seconds for all scenarios, the SotA approach got a mean value of 12.6 seconds. The SotA2 approach got the lowest MTTR with a mean value of 7 seconds for all scenarios. However, the SotP approach also reached interesting values for MTTR, with the second lowest accumulated mean values of 7.9 seconds. For all scenarios, we observed a weak correlation of the type of the scenario and the MTTR values. In this sense, little mean variation in the mean MTTR was observed for all scenarios.



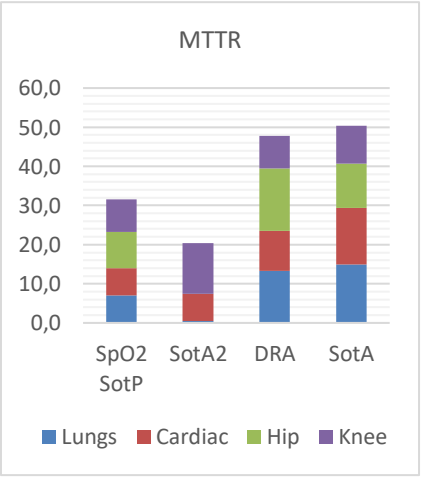


Figure 62 Total time for true positive service provision: (a) by approach, (b) by scenario

6.2.3 Discussion

Given the reported results of the simulations, we can draw some conclusions and establish relations between the scenarios and the evaluated approaches. We will present the argumentation and the discussions according to the research questions.

RQ1 – Accuracy

Effective situation assessment is a key element for improving the quality of care by increasing safety to avoid accidents or enhancing the patient's recovery through more precise treatment. In the section 6.2.1, we analyzed relevant evidence that the proposed DRA approach does indeed establish methods and techniques that leverage the accuracy of situation assessment compared with other approaches.

In Section 4.1, we presented our approach for deriving a more complete risk metric. Moreover, the risk assessment approach deals with system dynamicity challenges. This technique presents bold evidence in the results section 6.2.1 about its efficacy for the assessment of diverse tested scenarios. It got not only the highest accuracy factor, but also an impressive rate of sensitivity (true positives rate) and specificity (true negatives rate). We can infer that the completeness of the risk parameter elicitation process achieved by the approach fundamentally contributed to this level of precision and efficacy in the risk assessment. When analyzing the average number of parameters for the risk metric generated by the proposed approach, we observed a considerable difference for the other risk metrics. The proposed approach generated not only a higher number of parameters, but also considered the relevance of the parameters, which is missing in the other techniques.

Moreover, the proposed DRA approach got constant results for accuracy, sensitivity, and PatSitDect for the diverse scenarios. This also shows the capacity of the proposed risk metric to adapt to different scenarios and deal with variation in the different situations provided by the scenarios. Therefore, the DRA

approach can tune the appropriate risk metric for the current system state or context.

Given all the evidence observed in the results section, we can thus assure the efficacy of the proposed DRA approach for runtime risk assessment. As described in section 2, false alarms represent a key hazard for overdosage accidents in PCA treatment. By enhancing the accuracy and consequently the efficacy of the situation detection, the DRA approach mitigates false positives and reduces false alarms. This effect also increases the perceived actuation confidence of the system, which enhances the nursing care aspects of safety and team productivity.

RQ2 – Efficiency

An efficient risk assessment speeds up the detection of hazardous situations and can thus enable faster actuation by the caregivers in the case of PCA treatment to avoid accidents.

First of all, the sensitivity of the risk metric provided by the DRA approach is worth noting. In this context, sensitivity means the agility to analyze variances in the risk parameters within the scenarios. Sensitivity is directly related to either the relevant set of identified parameters or the parameter's weight in the risk metric. We have seen in Section 4 and in Section 5 that the DRA approach enhances the completeness and quantitative aspects of the risk metric and impacts the accuracy of the situational assessment. For example, we considered the context variables for the risk metric and the changes in the parameter's weight according to the system reconfiguration. Hence, we can assume that the superior performance of the DRA approach for hazardous situation detection is due to the enhanced sensitiveness of the risk metric as well as its accuracy regarding the assessment of the situation.

A second relevant aspect to consider in the efficiency results is the risk metric management performed by the DRA approach. Given the current system configuration, the risk assessment approach instantiates the proper risk metric. This capability enables fine-tuning of the risk metric for the current system and context state because of the flexibility to consider, for instance,

more parameters in case a new device is connected to the system. Moreover, this risk metric management enables changes in the relevance of the context and system parameters for the risk assessment. We can see evidence of the importance of risk metric adaptation in the results of the analysis of imprecise time (Figure 60a), which show that using the SotA approach, the imprecise time for the less critical scenarios was about twice as high compared to SotA2, which is adaptive.

Hence, the results provide evidence that the DRA approach yields enhanced efficiency in detecting hazardous situations compared to the other techniques. This efficiency supports faster reaction of the caregiver team, enabling them to take faster actions against respiration depression and thus mitigate the side effects of opioids or overdosage in the patient's body. Furthermore, such imprecise time values again increase the confidence in the system, which is a fundamental aspect for caregivers and helps to avoid false alarm fatigue. All in all, efficient risk assessment enables automated actuation for cooperative MCPS and thereby paves the way for the next generation of health systems with smart closed loops.

RQ3 – Treatment Availability

Treatment availability concerns another fundamental problem for PCA: underdosage. The higher the opioid availability, the more comfortable and rapid the recovery process. Hence, approaches should keep safety in mind without excessively restricting the treatment for the patients in order to fulfill the intent of the treatment.

In the results Section 6.2.1, we presented key evidence that treatment availability was enhanced by the proposed DRA approach. The highest TPU mean among the compared approaches for all scenarios indicates that it had the highest amount of time where it provided the service correctly, which was higher than 80%. As we recorded no false positive and false negative events for the DRA approach, we conclude that 16% of the total time spent during the situations was spent on true negative and reconfiguration time.

However, the MTBF and MTTR data also present an interesting aspect regarding service availability. The high accuracy of the DRA approach implies a smooth increase of service interruptions compared with the other approaches. This might sound contradictory at first glance, but note that MTBF and MTTR consider true negative and false negative events. As we measured no false negative events for the DRA approach, all the true negative events were responsible for the interrupt service provision of the approach. Therefore, considering MTTR and MTBF, treatment availability was not a sensible improvement for the DRA approach in spite of its accuracy.

In this sense, given the high TPU marks and struggling MTTR and MTBF values, we can still conclude that the DRA approach enhances treatment availability because of its high rate of time providing the service correctly and considering that the failure definition (interruption of the service) was right when the patient actually should not have the treatment.

7 Future Work

In this chapter, we will summarize and conclude this thesis, analyzing the claimed results and novel contributions as well as general benefits (cf. Section 7.1). Furthermore, we will discuss their limitations and open issues (cf. Section 7.2).

7.1 Conclusions and Limitations

In this thesis, we presented an approach for enabling dynamic risk assessment for MCPS and thus enhancing the accuracy, efficiency, and availability of the treatments supported by such systems. This work was motivated by the lack of risk assessment techniques that consider system and context dynamicity issues in a single risk assessment approach. This aspect is particularly relevant because, as discussed throughout this thesis, the lack of dynamic risk assessment techniques undermines both the runtime risk classification and risk control measures, due to the uncertainty about how the services will be realized at runtime or which context situation the system will face. Hence, the performance of the current risk assessment techniques has limited the adoption of system actuation features by care providers' institutions (state-of-the-Practice), which often focus only on alarms (as we have discussed in the section 2).

7.1.1 Specific contributions and limitations

The presented approach comprises two main constituents that focus on risk elicitation and analysis on the one hand and on risk classification and countermeasure specification on the other hand.

Contribution 1: *An adaptive risk analysis for DRA of cooperative medical CPS*:

The DRA approach for cooperative MCPS was developed to shift parts of the risk assessment to runtime. Therefore, we defined a set of guides and a metamodel to support safety engineers in the task of defining a dynamic risk model based on a Bayesian network in order to enable qualitative and quantitative assessment

at runtime. Considering the challenges for deriving an approach, we established: (i) a risk parameters elicitation metamodel; (ii) specification of a risk metric; and (iii) adaptive risk metric management.

The risk analysis approach (presented in Chapter 4) was developed to be performed in the early stages of the risk assessment process when elements such as hazards, hazardous situations, overall safety requirements, and abstract architecture(s) have already been created. Hence, the risk parameters elicitation metamodel aims to identify and classify relevant parameters regarding dynamic risk assessment. Moreover, the guide for deriving the risk metric based on the metamodel produce a tree-structured risk metric suitable, for example, for BBN implementation. The risk metric specification then enhances the process of deriving the risk metric for DRA. Finally, we provided guidelines and models for deriving adaptive risk metric management in order to deal with the challenges posed by system reconfigurations that might impact the risk metrics.

In a nutshell, we identified the following benefits during the validation process and the presentation of our approach to the safety community:

1. *Enhanced identification of relevant risk parameters*: The metamodel plays a key role in identifying a relevant set of information that needs to be monitored so that the risk metric can actually assess the current situational state. This model aggregates multidisciplinary information from different knowledge domains spread over several stakeholders. The classification takes into account different abstraction layers, the classification, and the relationships among the parameters. This structure can speed up the elicitation of the risk parameters that guide the safety engineers through the relevant concerns. Moreover, this also improves the completeness of the risk metric, since we can enhance the set of relevant risk parameters as well as their relationships to assess the risk. Therefore, this approach is not only relevant for DRA: we can potentially apply this metamodel in a regular risk assessment process to get the same benefits.

2. *Guide for defining the risk metric model*: The risk metric definition promotes a tree-structured risk metric to define a risk quantification approach. We built the risk metric using a Bayesian network, but it can potentially be implemented through other mathematical models as well. Regarding the quantitative aspects, the approach promotes an opportunity for reasoning together with the stakeholders about the weight of every single information node for the risk calculation, which, according to Nancy Leveson [65, 134, 135], is a key limitation of other approaches. Finally, the guidelines aim to build a risk metric for DRA of cooperative MCPS that introduces the aggregation of risk parameters not considered so far by the literature as we have presented in [130].

3. *Guide for specifying risk metric management*: Risk metric management enables adaptive risk monitoring. We defined a formal automata model to enable safety engineers to analyze and define how system reconfigurations will impact the risk metrics.

Regarding limitations of our approach, we found that the empirical definition of the quantitative values is the main issue of the approach. During the elicitation of the risk parameters, we gathered experience from medical experts, established literature, and standards to define the weights and aggregation functions of each parameter node in the risk model. However, these values are a matter of disagreement between medical experts and standards, due to subjective evaluation. Therefore, the final risk calculation might diverge when the approach is applied for different teams. For example, we observed various kinds of disagreement between the healthcare systems in Germany, Portugal, and Brazil. Hence, this can have a direct impact on the definition of the safety requirements due to the impact on the definition of the concept of risk.

We utilized Bayesian network as our output model for our risk analysis approach in order to enable an automatic model for assessing the risk at runtime. However, we understand that this could be enhanced by machine learning techniques. For example, we could enrich the CPTs of the Bayesian network with historical

data from a database such as PhysioNet, or we could implement the executable risk metric learning parameters from the historical data and train a knowledge base to assess the risk. However, until the end of this research, no related data was available, so we decided to capture the knowledge and uncertainty models from the medical experts with probabilistic BBN to assess the current risk of the situation.

Another limitation that we found is related to the completeness of the risk parameters derived from the metamodel. According to our experiments and the validation with medical experts (as reported in the Chapter 6), the approach can generate a risk metric capable of surpassing the accuracy, efficiency, and treatment availability of other state-of-the-art and state-of-the-practice solutions. However, we cannot assure that this set of parameters will always be the most complete set for all scenarios. In the medical domain, the scenarios might vary among widely different situations; for instance, cardiac surgery and knee surgery define different context demands, patients, and recovery treatments. Therefore, we need to expand the testing scenarios and include new hazards for different situations to enhance the completeness of the metamodel.

There are also limitations regarding the architectural assumptions for implementing the risk monitoring based on the system configurations. The approach assumes that the medical devices can expose some sort of safety guarantees, which can be based on runtime safety certification approaches [167, 195, 203]. This might be a severe limitation on the adoption in industry as few manufacturers employ such techniques in their products. Moreover, we considered a fixed number of abstract configurations for implementing the analyzed treatment supported by the MCPS. If a new abstract configuration emerges, the risk model needs to be redesigned to consider the safety guaratees, risk parameters, and aggregations for that new configuration.

Contribution 2: *Runtime risk evaluation and risk control specification*

We provided a classification structure for the risk level considering recommendations by standards regarding the severity

and probability of accidents. Therefore, we defined (i) a formal situation definition concept that supports safety engineers in reasoning about the situational risk at runtime; (ii) an abstract risk classification structure for runtime risk classification; and (iii) a specification for risk control measures. The latter considers alarms and warnings to alert the system environment about the risk level, proposes recommended actions and mechanisms to bring the system to the safe state, triggers adaptation mechanisms to derive a new configuration, and defines a certification process to achieve new guarantees according to the given configuration.

Benefits:

1. *Guide for reasoning about the situation analysis*: A key challenge for situational risk analysis is the fact that a complex set of parameters must be evaluated that contribute to the risk. The provided formal situational model empowers safety engineers with a language for specifying the relations between the relevant parameters for the risk. Hence, they can better manage the risk of missing important elements as well as their relations when assessing the current risk.

2. *Guide for practitioners for the derivation of risk classes based on standards*: Reasoning about risk classification for cooperative MCPS is challenging due to the system dynamcity. Context and system changes can impact on the way the risk is assessed, and this can indirectly impact on the way the risk will reach unacceptable and/or intolerable classes. Although the classical literature and standards do not consider system changes, the proposed approach enables reasoning about risk acceptance for such systems assuming the DRA approach.

3. *Countermeasure allocation and classification (based on the literature and an analysis for dynamic and cooperative systems)*: The standards and the modern literature on safety requirements definition allow only a limited and well-defined set of system changes. Our approach assumes such limitations, but enables going one step further because we permit runtime evaluation of the

impact of countermeasures on the whole situational risk. Likewise, it allows some safety requirements to dynamically become mandatory given the current risk.

4. *Architectural compliance and implementation*: The final step of this contribution extends the reference architecture ASTM F2761 – 09(2013)[72] from the standards to realize the risk management algorithm that enables the adaptive risk monitoring model. This is a relevant benefit for the safety community as we were able to extend the model for handling safety issues, which was not considered by the architectural reference. Now industry and academia can establish their solutions over the proposed architectural model. Moreover, adaptive risk monitoring plays a fundamental role in the DRA technique for cooperative MCPS.

A key limitation of this contribution is how to deal with the subjectivism behind the definition of safety requirements. The risk classification technique for DRA of cooperative MCPS still relies on the evaluation of safety engineers for establishing the risk classes. During the first validation experiment, we had several interactions with medical experts to define proper risk classes. In spite of the refined guide provided by the approach, there is still room for uncertainty and disagreement, which can either undermine the whole risk assessment approach and thus limit the system availability with conservative risk class values or lead to ineffective risk control actions being taken.

A second limitation of the approach is related to how to determine the ranges of the risk classes. We defined fixed values for the transitions between the risk classes. This approach might need some further refinement due to the risk curve behavior assessed by the risk monitor. For instance, the risk behavior could be defined by a hysteresis effect for state transitions. This could enhance the precision of the risk classification as well as the proper activation of the risk measures.

The third limitation is related to the limited set of safety requirements that our implementation model uses. We analyzed the literature classification and identified which type of measures

can actually play an effective role for cooperative MCPS considering their dynamicity. However, there is still room for improvement in this area, for example regarding issues such as how to assess the impact of the human in the loop, or regarding major architectural changes in the system (beyond the predefined abstract architecture configurations). Hence, we understand that the space for safety requirements definition is feasibly practical, but it could be more flexible and consider a wider range of countermeasures.

7.2 Future work

Machine learning techniques could provide new research directions for the challenge of how to establish risk levels. Assuming the popularization of such clinical applications of MCPS, hospitals and care providers could be compelled to provide support with their databases in order to enable the application of data-intensive techniques. Such techniques could learn patterns and build up more fine-grained risk models, considering the complexity of the human body's behavior in interaction with each type of opioid and the respective clinical situation. Moreover, aspects related to the formalization of confidence in the risk metric also need to be addressed by such models. Hence, we assume that such techniques could enhance the accuracy and efficiency of the risk metrics.

A second fundamental aspect for the evolution of the risk meta-model would be the application of the model and the guidelines to different scenarios and other domains. We believe that the proposed meta-model can contribute to the establishment of risk metrics for different domains. For example, although we have considered correlated elements with autonomy, it could impact the definition of new parameters (or abstraction layers) for the metamodel. Therefore, the application of the meta-model over different areas could also bring new aspects to it that have not been considered so far.

A third aspect for future work is related to safety argumentation and runtime assurance. The main question refers to how to provide enough evidence for developing safety arguments that assure the safety of the system for all system states. The risk

assessment model could provide some piece of information with some extent of confidence to support a dynamic or modular argument that could be built on-the-fly [29, 193, 240].

A fundamental aspect that has not been the focus of this thesis is security. Safety and security are increasingly required to be integrated into system development and assurance [76, 143, 205]. We understand that a key element for deriving an integrated approach would be the risk assessment model. We are aware of some works that propose a risk metric for security [9, 10, 183, 197, 200, 250]; however, the integration between the two methodologies is still an open issue for industry and research. Therefore, the contributions of this thesis enable the establishment of a suitable environment for research and development of novel techniques in this direction.

8 References

1. (ECRI), E.C.R.I.: Top 10 Health Technology Hazards for 2017. (2016).

2. (ECRI), E.C.R.I.: Top 10 Health Technology Hazards for 2020. (2020).

3. Abimbola, M. et al.: Safety and risk analysis of managed pressure drilling operation using Bayesian network. Saf. Sci. 76, (2015).

4. Acharya, S. et al.: Pulse Oximeter Signal Modeling and Fusion for Hypoxia Monitoring. Inf. Fusion (FUSION), 2014. (2014).

5. Adler, R. et al.: Engineering Dynamic Adaptation for Achieving Cost-Efficient Resilience in Software-Intensive Embedded Systems. In: 2010 15th IEEE International Conference on Engineering of Complex Computer Systems. pp. 21–30 IEEE (2010).

6. Adler, R. et al.: Safety Engineering for Autonomous Vehicles. In: 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W). pp. 200–205 IEEE (2016).

7. Adler, R., Kemmann, S.: Towards safe autonomic driving Enhancing traditional hazard and risk analysis. In: Berns, Karsten; Schindler, Christian; Dreßler, K.; Jörg, B.; Kalmar, Ralf; Zolynski, G. (ed.) Proceedings of the 3rd Commercial Vehicle Technology Symposium (CVT 2014), ISBN: 978-3-8440-2573-6. pp. 2–12 Aachen Shaker Verlag, Kaiserslautern (2014).

8. Allenby, K., Kelly, T.: Deriving safety requirements using scenarios. In: Proceedings Fifth IEEE International Symposium on Requirements Engineering. pp. 228–235 IEEE Comput. Soc (2001).

9. Amorim, T. et al.: Multidirectional modular conditional safety certificates. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 9338, 357–368 (2015).

10. Amorim, T. et al.: Systematic Pattern Approach for Safety and Security Co-engineering in the Automotive Domain. Presented at the (2017).

11. Andreu-Perez, J. et al.: Big Data for Health. IEEE J. Biomed. Heal. Informatics. 19, 4, 1193–1208 (2015).

12. Arney, D. et al.: Design pillars for medical cyber-physical system middleware. 5th Med. Cyber-Physical Syst. Work. Device Interoperability, Safety, Secur. Assur. MCPS 2014. 36, 124–132 (2014).

13. Arney, D. et al.: Toward patient safety in closed-loop medical device systems. In: Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems - ICCPS '10. p. 139 ACM Press, New York, New York, USA (2010).

14. Avizienis, A. et al.: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Trans. Dependable Secur. Comput. 1, 1, 11–33 (2004).

15. Badue, C. et al.: Self-driving cars: A survey. Expert Syst. Appl. 165, 113816 (2021).

16. Barbosa, P.E.S. et al.: RE4CH : Requirements Engineering for Connected Health.

17. Berkenstadt, H. et al.: An evaluation of the Integrated Pulmonary Index (IPI) for the detection of respiratory events in sedated patients undergoing colonoscopy. J. Clin. Monit. Comput. 26, 3, 177–181 (2012).

18. Berner, E.S.: Clinical Decision Support Systems. Springer International Publishing, Cham (2016).

19. Berner, E.S. et al.: Performance of Four Computer-Based Diagnostic Systems. N. Engl. J. Med. 330, 25, 1792–1796 (1994).

20. Berner, E.S., La Lande, T.J.: Overview of Clinical Decision Support Systems. Presented at the (2007).

21. Biradar, H.B., Shabadi, L.: Review on IOT based multidisciplinary models for smart farming. In: RTEICT 2017 - 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, Proceedings. pp. 1923–1926 Institute of Electrical and Electronics Engineers Inc. (2017).

22. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2011).

23. Bitterman, D.S. et al.: Approaching autonomy in medical artificial intelligence, https://www.faa.gov/aircraft/, (2020).

24. Bondavalli, A. et al.: Cyber-physical systems of systems : foundations -- a conceptual model and some derivations: the AMADEOS legacy.

25. Brito, M., Griffiths, G.: A Bayesian approach for predicting risk of autonomous underwater vehicle loss during their missions. Reliab. Eng. Syst. Saf. 146, 55–67 (2016).

26. Broy, M. et al.: Cyber-physical systems: Imminent challenges. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 7539 LNCS, 1–28 (2012).

27. Broy, M.: Engineering Cyber-Physical Systems: Challenges and Foundations. Complex Syst. Des. Manag. 1–13 (2013).

28. Cai, B. et al.: Multi-source information fusion based fault diagnosis of ground-source heat pump using Bayesian network. Appl. Energy. 114, 1–9 (2014).

29. Calinescu, R. et al.: Engineering trustworthy self-adaptive software with dynamic assurance cases. IEEE Trans. Softw. Eng. 44, 11, (2018).

30. Cano, A. et al.: A Method for Integrating Expert Knowledge When Learning Bayesian Networks From Data. IEEE Trans. Syst. Man, Cybern. Part B. 41, 5, 1382–1394 (2011).

31. Carter, J.H.: Design and Implementation Issues. In: Clinical Decision Support Systems. pp. 64–98 Springer New York (2007).

32. CCPS: Layer of Protection Analysis: Simplified Process Risk Assessment. Wiley (2001).

33. Celdran, A.H. et al.: ICE++: Improving security, QoS, and high availability of medical cyber-physical systems through mobile edge computing. In: 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services, Healthcom 2018. pp. 1–8 IEEE, Ostrava (2018).

34. Challa, S., Koks, D.: Bayesian and Dempster-Shafer fusion. Sadhana. 29, 2, 145–174 (2004).

35. Chen, S. et al.: Data-driven Adaptive Safety Monitoring using Virtual Subjects in Medical Cyber-Physical Systems: A Glucose Control Case Study Recommended Citation "Data-driven Adaptive Safety Monitoring using Virtual Subjects in Data-driven Adaptive Safety Monitorin. J. Comput. Sci. Eng. J. Comput. Sci. Eng. J. Comput. Sci. Eng. J. Comput. Sci. Eng. 10, 3, 75–84 (2016).

36. Cicotti, G., Coronato, A.: Towards a Probabilistic Model Checking-based approach for Medical Device Risk Assessment. 2015 IEEE Int. Symp. Med. Meas. Appl. MeMeA 2015 - Proc. 180–185 (2015).

37. Clifton A. Ericson: Hazard and Operability Analysis. In: Hazard Analysis Techniques for System Safety. pp. 365–381 John Wiley & Sons, Inc., Hoboken, NJ, USA (2005).

38. Cohen, M.R., Smetzer, J.L.: ISMP Medication Error Report Analysis - Fatal Patient-Controlled Anesthesia Adverse Events; Name Confusion with New Cancer Drugs; Medication Safety Officer Group to Become a Part of ISMP. Hosp. Pharm. 48, 9, 715–724 (2013).

39. Constantinou, A.C. et al.: Integrating Expert Knowledge with Data in Bayesian Networks: Preserving Data-Driven Expectations when the Expert Variables Remain Unobserved. Expert Syst. Appl. 56, 197–208 (2016).

40. Cvach, M. et al.: Clinical Alarms and the Impact on Patient Safety. Clin. Eng. 1–8 (2006).

41. Daimler AG: Autonomous Driving. (2016).

42. Das, J. et al.: Pulse oximeter accuracy and precision at fifi ve different sensor locations in infants and children with cyanotic heart disease. Indian J. Anaesth. 54, 6, 531–534 (2010).

43. Dayan, P.S. et al.: Use of Traumatic Brain Injury Prediction Rules With Clinical Decision Support. Pediatrics. 139, 4, e20162709 (2017).

44. Denney, E. et al.: Perspectives on software safety case development for unmanned aircraft. Proc. Int. Conf. Dependable Syst. Networks. (2012).

45. Dervishi, A.: Fuzzy risk stratification and risk assessment model for clinical monitoring in the ICU. Comput. Biol. Med. 87, May, 169–178 (2017).

46. Dey, N. et al.: Medical cyber-physical systems: A survey, (2018).

47. Dumitrache, I. et al.: A Cyber Physical Systems Approach for Agricultural Enterprise and Sustainable Agriculture. In: Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017. pp. 477–484 Institute of Electrical and Electronics Engineers Inc. (2017).

48. Dunjó, J. et al.: Hazard and operability (HAZOP) analysis. A literature review. J. Hazard. Mater. 173, 1–3, 19–32 (2010).

49. Edworthy, J., Hellier, E.: Fewer but better auditory alarms will improve patient safety. Qual. Saf. Health Care. 14, 3, 212–5 (2005).

50. Elisa Carcereri de Oliveira, A.I. et al.: Alarm fatigue and the implications for

patient safety. Rev Bras Enferm [Internet]. 71, 6, 3035–3075 (2018).

51. Emergency Care Research Institute (ECRI): Top 10 Health Technology Hazards for 2014. (2014).

52. Emergency Care Research Institute (ECRI): Top 10 Health Technology Hazards for 2019. (2018).

53. Ericson, C. a.: Event Tree Analysis. Hazard Anal. Tech. Syst. Saf. 223–234 (2005).

54. Ericson, C.A.: Hazard Analysis Techniques for System Safety. John Wiley & Sons (2005).

55. Erturkmen, G.B.L. et al.: Personalised care plan management utilizing guideline-driven clinical decision support systems. In: Studies in Health Technology and Informatics. pp. 750–754 IOS Press (2018).

56. Von Essen, C., Giannakopoulou, D.: Analyzing the next generation airborne collision avoidance system. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 620–635 Springer Verlag (2014).

57. Fenelon, P., Hebbron, B.: Applying HAZOP to software engineering models. Risk Manag. Crit. Prot. Syst. …. (1994).

58. Fenton, N., Neil, M.: Risk assessment and decision anlysis with Bayesian networks. (2013).

59. Fernandes, M. et al.: Clinical Decision Support Systems for Triage in the Emergency Department using Intelligent Systems: a Review. Artif. Intell. Med. 102, August 2019, 101762 (2020).

60. Feth, P. et al.: A Conceptual Safety Supervisor Definition and Evaluation Framework for Autonomous Systems. In: Tonetta S., Schoitsch E., B.F. (eds) (ed.) Computer Safety, Reliability, and Security. SAFECOMP 2017. pp. 135–148 (2017).

61. Feth, P. et al.: A Conceptual Safety Supervisor Definition and Evaluation Framework for Autonomous Systems. Comput. Safety, Reliab. Secur. (2017).

62. Feth, P. et al.: A Context-Aware, Confidence-Disclosing and Fail-Operational Dynamic Risk Assessment Architecture. In: Proceedings - 2018 14th European Dependable Computing Conference, EDCC 2018. (2018).

63. Feth, P. et al.: Multi-Aspect Safety Engineering for Highly Automated Driving.

64. Feussner, H. et al.: Surgery 4.0. Heal. 4.0 How Virtualization Big Data are Revolutionizing Healthc. (2017).

65. Fleming, C.H. et al.: Safety assurance in NextGen and complex transportation systems. Saf. Sci. 55, 173–187 (2013).

66. Frederickson, A.A.: The Layer of Protection Analysis (LOPA) method Look for best practices and guidelines on how to use the LOPA method as an alternative to mitigate risks . 1997, 1–9 (2002).

67. Friedman, C.P., Wyatt, J.: Evaluation methods in biomedical informatics. Springer-Verlag New York (2006).

68. Gagniuc, P.A.: Markov Chains: From Theory to Implementation and Experimentation. Wiley (2017).

69. Gasser, T. M.; Arzt, C.; Ayoubi, M.; Bartels, A.; Bürkle, L.; Eier, J.; Flemisch, F.; Häcker, D.; Hesse, T.; Huber, W.; Lotz, C.; Maurer, M.; Ruth-Schumacher, S.; Schwarz, J.; Vogt, W.: Rechtsfolgen zunehmender Fahrzeugautomatisierung [The legal consequences of increasing vehicle automation]. , Bergisch Gladbach (2012).

70. Gatouillat, A. et al.: Internet of Medical Things: A Review of Recent Contributions Dealing with Cyber-Physical Systems in Medicine. IEEE Internet Things J. 5, 5, 3810–3822 (2018).

71. Gempton, N. et al.: Autonomous control in military logistics vehicles: Trust and safety analysis. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 8020 LNAI, PART 2, 253–262 (2013).

72. Goldman, J.M.: Medical Devices and Medical Systems - Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) - Part 1: General requirements and conceptual model. (2009).

73. Grigoriadis, N. et al.: Health 4.0: The case of multiple sclerosis. In: 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services, Healthcom 2016. Institute of Electrical and Electronics Engineers Inc. (2016).

74. Gusrialdi, A. et al.: Competitive Interaction Design of Cooperative Systems Against Attacks. IEEE Trans. Automat. Contr. 63, 9, 3159–3166 (2018).

75. Haddadin, S. et al.: Towards the robotic co-worker. In: Springer Tracts in

Advanced Robotics. pp. 261–282 Springer, Berlin, Heidelberg (2011).

76. Halperin, D. et al.: Security and Privacy for Implantable Medical Devices. IEEE Pervasive Comput. 7, 1, 30–39 (2008).

77. Hatcliff, J. et al.: Rationale and Architecture Principles for Medical Application Platforms. 2012 IEEE/ACM Third Int. Conf. Cyber-Physical Syst. 3–12 (2012).

78. Health and Safety Executive: ALARP Guidance: UK Health and Safety Executive, https://www.hse.gov.uk/managing/theory/index.htm.

79. Hegde, J., Rokseth, B.: Applications of machine learning methods for engineering risk assessment – A review, (2020).

80. Hilmi Ismail, Z., Sariff, N.: A Survey and Analysis of Cooperative Multi-Agent Robot Systems: Challenges and Directions. In: Applications of Mobile Robots. IntechOpen (2019).

81. Hollnagel, E.: Risk + barriers = safety? Saf. Sci. 46, 2, 221–229 (2008).

82. Holmes, D.E., Jain, L.C.: Introduction to Bayesian Networks. In: Innovations in Bayesian Networks. pp. 1–5 Springer Berlin Heidelberg, Berlin, Heidelberg (2008).

83. Http://Www.Faa.Gov/: Chapter 7: Integrated System Hazard Analysis. FAA Syst. Saf. Handb. 1–18 (2000).

84. Hu, Y.-J. et al.: Decision tree-based learning to predict patient controlled analgesia consumption and readjustment. BMC Med. Inform. Decis. Mak. 12, 1, 131 (2012).

85. Hudcova, J. et al.: Patient controlled opioid analgesia versus conventional opioid analgesia for postoperative pain. Cochrane database Syst. Rev. 4, CD003348 (2006).

86. Ibañez-Guzmán, J. et al.: Autonomous Driving: Context and State-of-the-Art. Handb. Intell. Veh. 1271–1310 (2012).

87. IEC: BS IEC 61882:2001, Hazard and Operability Studies (HAZOP Studies)—Application Guide. (2001).

88. IEC: Functional safety of electrical / electronic / programmable electronic safety-related systems - part 7: Overview of techniques and measure. (2010).

89. IEC: IEC 61025:2006-12 - Fault Tree Analysis. International Electrotechnical

Commission (2006).

90. IEC: IEC 62304: Medical device software -- Software life cycle processes. (2006).

91. IEC: IEC EN 61508-5:2010 Functional safety of electrical/ electronic/programmable electronic safety related systems - Part 5: Examples of methods for the determination of safety integrity levels. (2010).

92. IEC: Medical electrical equipment — Part 1-8: General requirements for basic safety and essential performance — Collateral standard: General requirements, tests and guidance for alarm systems in medical electrical equipment and medical electrical systems. (2006).

93. Insup Lee et al.: Challenges and Research Directions in Medical Cyber-Physical Systems. Proc. IEEE. 100, 1, 75–90 (2012).

94. Irfan, M., Ahmad, N.: Internet of medical things: Architectural model, motivational factors and impediments. In: 2018 15th Learning and Technology Conference, L and T 2018. pp. 6–13 Institute of Electrical and Electronics Engineers Inc. (2018).

95. ISO: ISO 14971:2019 - Medical devices — Application of risk management to medical devices. (2019).

96. ISO, IEC: ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. (2011).

97. Ivanov, I.I.: Exploring business - IT nexus: Make the most of IT-enabled capabilities. Lect. Notes Bus. Inf. Process. 257, 152–170 (2016).

98. Ivanov, R. et al.: Context-Aware Detection in Medical Cyber-Physical Systems. In: Proceedings - 9th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2018. pp. 232–241 Institute of Electrical and Electronics Engineers Inc. (2018).

99. Ivanov, R. et al.: Prediction of Critical Pulmonary Shunts in Infants. IEEE Trans. Control Syst. Technol. 24, 6, 1936–1952 (2016).

100. Jahan, F. et al.: Security Modeling of Autonomous Systems. ACM Comput. Surv. 52, 5, 1–34 (2019).

101. Jamshidi, M.: System of Systems Engineering. John Wiley & Sons, Inc., Hoboken, NJ, USA (2008).

102. Jensen, F. V.: An introduction to Bayesian networks. Springer (1996).

103. Jezewski, J. et al.: Towards a medical cyber-physical system for home telecare of high-risk pregnancy. IFAC-PapersOnLine. 48, 4, 466–473 (2015).

104. Jia, X. et al.: A novel semi-supervised deep learning framework for affective state recognition on EEG signals. In: Proceedings - IEEE 14th International Conference on Bioinformatics and Bioengineering, BIBE 2014. pp. 30–37 Institute of Electrical and Electronics Engineers Inc. (2014).

105. Jiang, Y. et al.: A Self-Adaptively Evolutionary Screening Approach for Sepsis Patient. Proc. - IEEE Symp. Comput. Med. Syst. 2016-Augus, 60–65 (2016).

106. Jiang, Y. et al.: Sepsis Patient Detection and Monitor Based on Auto-BN. J. Med. Syst. 40, 4, (2016).

107. Jin, Z. et al.: CarePre: An Intelligent Clinical Decision Assistance System. 1, 1, 1–20 (2018).

108. Johansson, R. et al.: Functional Safety and Evolvable Architectures for Autonomy. Autom. Driv. - Safer More Effic. Futur. Driv. (2016).

109. John Rushby: Kernels for Safety? In: Anderson, T. (ed.) Safe and Secure Computing Systems. pp. 210–220 Blackwell Scientific Publications, Glasgow (1989).

110. Kazemi, R. et al.: A Hybrid Methodology for Modeling Risk of Adverse Events in Complex Health-Care Settings. Risk Anal. 37, 3, 421–440 (2017).

111. Kelly, D.L., Smith, C.L.: Bayesian inference in probabilistic risk assessment—The current state of the art. Reliab. Eng. Syst. Saf. 94, 2, 628–643 (2009).

112. Kemmann, S.: SAHARA A Structured Approach for Hazard Analysis and Risk Assessments. Technische Universität Kaiserslautern (2015).

113. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer (Long. Beach. Calif). 36, 1, 41–50 (2003).

114. Khakzad, N. et al.: Dynamic risk analysis using bow-tie approach. Reliab. Eng. Syst. Saf. 104, 36–44 (2012).

115. Kim, H. et al.: Reliability data update using condition monitoring and prognostics in probabilistic safety assessment. Nucl. Eng. Technol. 47, 2, 204–211 (2015).

116. King, A. et al.: Evaluation of a smart alarm for intensive care using clinical data. Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf. 2012, 166–9 (2012).

117. King, A. et al.: Prototyping closed loop physiologic control with the medical device coordination framework. In: Proceedings of the 2010 ICSE Workshop on Software Engineering in Health Care - SEHC '10. pp. 1–11 ACM Press, New York, New York, USA (2010).

118. King, A.L. et al.: Towards assurance for plug & Play medical systems. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 9337, 228–242 (2015).

119. Klein, P.: The safety-bag expert system in the electronic railway interlocking system elektra. Expert Syst. Appl. 3, 4, (1991).

120. Klerkx, L. et al.: A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda, (2019).

121. Kochenderfer, M.J. et al.: Next-Generation Airborne Collision Avoidance System. , Massachusetts (2013).

122. Krupitzer, C. et al.: A survey on engineering approaches for self-adaptive systems. Pervasive Mob. Comput. 17, PB, 184–206 (2015).

123. Kuramoto, T.: Risk Monitoring for Nuclear Power Plant Applications Using Probabilistic Risk Assessment. In: Progress of Nuclear Safety for Symbiosis and Sustainability. pp. 145–151 Springer Japan, Tokyo (2014).

124. Kurd, Z. et al.: Establishing a framework for dynamic risk management in "intelligent" aero-engine control. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 326–341 (2009).

125. Kurd, Z. et al.: Towards establishing a self-management architecture for dynamic risk management in "Intelligent" aero-engine control. IET Conf. Publ. 2009, 555 CP, (2009).

126. Larson, B. et al.: Requirements specification for apps in medical application platforms. 2012 4th Int. Work. Softw. Eng. Heal. Care. 26–32 (2012).

127. Lasi, H. et al.: Industry 4.0. Bus. Inf. Syst. Eng. 6, 4, 239–242 (2014).

128. Lee, E.A.: CPS foundations. In: Proceedings of the 47th Design Automation Conference on - DAC '10. p. 737 ACM Press, New York, New York, USA

(2010).

129. Lee, I. et al.: Challenges and Research Directions in Medical Cyber–Physical Systems. Proc. …. 100, 1, 75–90 (2012).

130. Leite, F.L. et al.: Dynamic risk assessment enabling automated interventions for medical cyber-physical systems. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 11698 LNCS, 216–231 (2019).

131. Leite, F.L. et al.: Dynamic risk management for cooperative autonomous medical cyber-physical systems. In: Gallina B., Skavhaug A., Schoitsch E., B.F. (ed.) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 126–138 Springer, Cham (2018).

132. Leite, F.L. et al.: Safety Assurance for Autonomous and Collaborative Medical Cyber-Physical Systems. In: Tonetta S., Schoitsch E., B.F. (ed.) Computer Safety, Reliability, and Security. SAFECOMP 2017. pp. 237–248 Springer, Cham (2017).

133. De Lemos, R. et al.: Software engineering for self-adaptive systems: A second research roadmap. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 7475 LNCS, 1–32 (2013).

134. Leveson, N.: A systems approach to risk management through leading safety indicators. Reliab. Eng. Syst. Saf. 136, 17–34 (2015).

135. Leveson, N.G.: Engineering a Safer World: Systems Thinking Applied to Safety. (2011).

136. Li, M. et al.: Dynamic risk assessment in healthcare based on Bayesian approach. Reliab. Eng. Syst. Saf. 189, 327–334 (2019).

137. Li, T. et al.: Towards context-aware medical cyber-physical systems: design methodology and a case study. Cyber-Physical Syst. 1, 1, 5–23 (2015).

138. Li, W. et al.: Towards false alarm reduction using fuzzy if-Then rules for medical cyber physical systems. IEEE Access. 6, 6530–6539 (2018).

139. Liggesmeyer, P.: Software-Qualität. Springer Spektrum (2009).

140. Lochner, C.M. et al.: All-organic optoelectronic sensor for pulse oximetry. Nat. Commun. 5, 1, 1–7 (2014).

141. Louis, S.: Functional Hazard Analysis (FHA) Methodology Tutorial International System Safety Training Symposium. (2014).

142. Lynn, L.A., Curry, J.P.: Patterns of unexpected in-hospital deaths: a root cause analysis. Patient Saf. Surg. 5, 1, 3 (2011).

143. Macher, G. et al.: Towards dependability engineering of cooperative automotive cyber-physical systems. In: Communications in Computer and Information Science. pp. 205–215 (2017).

144. Machin, M. et al.: SMOF: A Safety Monitoring Framework for Autonomous Systems. IEEE Trans. Syst. Man, Cybern. Syst. 99, 1–14 (2016).

145. Machin, M. et al.: Specifying Safety Monitors for Autonomous Systems Using Model-Checking. Presented at the (2014).

146. Maddox, R.R. et al.: Clinical experience with patient-controlled analgesia using continuous respiratory monitoring and a smart infusion system. Am. J. Heal. Pharm. 63, 2, 157–164 (2006).

147. Maddox, R.R. et al.: Continuous Respiratory Monitoring and a “Smart” Infusion System Improve Safety of Patient-Controlled Analgesia in the Postoperative Period. Agency for Healthcare Research and Quality (US), Rockville, MD, USA (2008).

148. Maglogiannis, I. et al.: Risk analysis of a patient monitoring system using Bayesian Network modeling. J. Biomed. Inform. 39, 6, 637–647 (2006).

149. Maier, M.W.: Architecting principles for systems-of-systems. Syst. Eng. 1, 4, 267–284 (1998).

150. Manning, C.D. et al.: Introduction to Information Retrieval. Cambridge University Press (2008).

151. Marakas, G.M.: Decision support system. Prentice Hall (2002).

152. Markowski, A.S., Kotynia, A.: “Bow-tie” model in layer of protection analysis. Process Saf. Environ. Prot. 89, 4, 205–213 (2011).

153. Marrubini, M.B.: Classifications of coma. Intensive Care Med. 10, 5, 217–226 (1984).

154. Masson, L. et al.: Synthesis of safety rules for active monitoring: Application to an airport light measurement robot. In: Proceedings - 2017 1st IEEE International Conference on Robotic Computing, IRC 2017. (2017).

155. McCarter, T. et al.: Capnography monitoring enhances safety of postoperative patient-controlled analgesia. Am. Heal. drug benefits. 1, 5, 28–35 (2008).

156. McCarter, T. et al.: Capnography monitoring enhances safety of postoperative patient-controlled analgesia. Am. Heal. Drug Benefits. 1, June, 28–35 (2008).

157. McDermid, J.A., Pumfrey, D.J.: A development of hazard analysis to aid software design. In: Proceedings of COMPASS'94 - 1994 IEEE 9th Annual Conference on Computer Assurance. pp. 17–25 IEEE.

158. McGrath, S.P. et al.: Improving Patient Safety and Clinician Workflow in the General Care Setting With Enhanced Surveillance Monitoring. IEEE J. Biomed. Heal. Informatics. 23, 2, 857–866 (2019).

159. McHeick, H. et al.: Survey of health care context models and prototyping of healthcare context framework. In: SCSC '16 Proceedings of the Summer Computer Simulation Conference. pp. 60:1--60:8 Society for Computer Simulation International San Diego, CA, USA ©2016, Montreal, Quebec, Canada (2016).

160. McNicol, E.D. et al.: Patient controlled opioid analgesia versus non-patient controlled opioid analgesia for postoperative pain. Cochrane database Syst. Rev. 6, CD003348 (2015).

161. Meisenberg, B. et al.: Implementation of solutions to reduce opioid-induced oversedation and respiratory depression. Am. J. Heal. Pharm. (2016).

162. Mekki-Mokhtar, A. et al.: Elicitation of Executable Safety Rules for Critical Autonomous Systems. 6th Eur. Congr. Embed. Real-Time Softw. Syst. (2012).

163. Mostafa, S.A. et al.: Adjustable autonomy: a systematic literature review. Artif. Intell. Rev. 51, 2, 149–186 (2019).

164. Mosterman, P.J., Zander, J.: Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems. Softw. Syst. Model. 15, 1, 5–16 (2016).

165. Mosterman, P.J., Zander, J.: Industry 4.0 as a Cyber-Physical System study. Softw. Syst. Model. 15, 1, 17–29 (2016).

166. Muccini, H. et al.: Self-adaptation for cyber-physical systems. In: Proceedings of the 11th International Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '16. pp. 75–81 ACM Press, New York, New York, USA (2016).

167. Müller, S., Liggesmeyer, P.: A motion certification concept to evaluate operational safety and optimizing operating parameters at runtime. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 9338, 156–166 (2015).

168. Müller, S., Liggesmeyer, P.: Systematic Composition of Services from Distributed Systems for Highly Dynamic Collaboration Processes. In: Computer Safety, Reliability, and Security. pp. 225–236 (2017).

169. Murphey, R.: An Introduction to Collective and Cooperative Systems. In: Cooperative Control and Optimization. pp. 171–197 Kluwer Academic Publishers (2005).

170. Nancy G. Leveson: Safeware: System Safety and Computers. Addison-Wesley Professional (1995).

171. Nielsen, C.B. et al.: Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. ACM Comput. Surv. 48, 2, 1–41 (2015).

172. Paine, C. et al.: Systematic Review of Physiologic Monitor Alarm Characteristics and Pragmatic Interventions to Reduce Alarm Frequency. J. Hosp. Med. 11 2, 136–144 (2016).

173. Pajic, M. et al.: Model-Driven Safety Analysis of Closed-Loop Medical Systems. IEEE Trans. Ind. informatics. 10, 1, 3–16 (2012).

174. De Paola, A. et al.: Multi-sensor fusion through adaptive bayesian networks. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 6934 LNAI, 360–371 (2011).

175. Papadopoulos, Y. et al.: Analysis and synthesis of the behaviour of complex programmable electronic systems in conditions of failure. Reliab. Eng. Syst. Saf. 71, 3, 229–247 (2001).

176. Patel, R. et al.: Smart Healthcare System Using IoT. In: Lecture Notes in Electrical Engineering. pp. 149–156 Springer (2020).

177. Perreault, L.E. and Metzger, J.B.: A pragmatic framework for understanding clinical decision support. J. Healthc. Inf. Manag. 13, 2, 5–21 (1999).

178. Petri, D.C.A.: Communication with Automata. (1966).

179. Plourde, J. et al.: OpenICE: An open, interoperable platform for medical cyber-physical systems. In: 2014 ACM/IEEE International Conference on Cyber-

Physical Systems (ICCPS). pp. 221–221 IEEE (2014).

180. Public, U.S.C. on the I.M. for A. of R. to: Risk Assessment in the Federal Government. National Academies Press, Washington, D.C. (1983).

181. Purdy, G.: ISO 31000:2009 - Setting a new standard for risk management: Perspective. Risk Anal. 30, 6, 881–886 (2010).

182. Rajamani, R., Zhu, C.: Semi-autonomous adaptive cruise control systems. IEEE Trans. Veh. Technol. 51, 5, 1186–1192 (2002).

183. Rao, A. et al.: COMPOSITE RISK MODELING FOR AUTOMATED. In: Rozenblit, J.W. and Sametinger, J. (eds.) Proceeding MSM '17 Proceedings of the Symposium on Modeling and Simulation in Medicine Article No. 9. Society for Computer Simulation International San Diego, CA, USA ©2017, Virginia Beach, Virginia (2017).

184. Rausand, M.: Preliminary Hazard Analysis. October. 1–36 (2005).

185. Ravi, D. et al.: Deep Learning for Health Informatics. IEEE J. Biomed. Heal. Informatics. 21, 1, 4–21 (2017).

186. Roderick, S. et al.: The ranger robotic satellite servicer and its autonomous software-based safety system, (2004).

187. Roederer, A. et al.: Clinical Decision Support for Integrated Cyber-Physical Systems : A Mixed Methods Approach. Proc. 3rd ACM Int. Heal. Informatics Symp. (2012).

188. Romero, J.A. et al.: Robotics and Road Transportation : A Review. Intell. Robot. Appl. 467–478 (2014).

189. Ross, T.J. (University of N.M.: Fuzzy logic with engineering applications. (2010).

190. Rubí, J.N.S., Gondim, P.R. de L.: Interoperable Internet of Medical Things platform for e-Health applications. Int. J. Distrib. Sens. Networks. 16, 1, 155014771988959 (2020).

191. de Ruijter, A., Guldenmund, F.: The bowtie method: A review. Saf. Sci. (2015).

192. Ruijters, E., Stoelinga, M.: Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. Comput. Sci. Rev. 15–16, 29–62 (2015).

193. Rushby, J.: A Safety-Case Approach for Certifying Adaptive Systems. AIAA

Infotech@aerosp. Conf. 1–16 (2009).

194. Rushby, J.: Just-in-Time Certification. In: 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007). pp. 15–24 IEEE (2007).

195. Rushby, J.: Runtime Certification. Runtime Verif. 21–35 (2008).

196. Rushby, J.: Runtime Verification. In: Leucker, M. (ed.) Eighth Workshop on Runtime Verification: RV08. pp. 21–35 Springer Berlin Heidelberg, Berlin, Heidelberg (2008).

197. Sadvandi, S. et al.: Safety and security interdependencies in complex systems and SoS: Challenges and perspectives. Proc. 2nd Int. Conf. Complex Syst. Des. Manag. CSDM 2011. 229–241 (2011).

198. SAE International: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles J3016, (2018).

199. Sandberg, A. et al.: Model-based safety engineering of interdependent functions in automotive vehicles using EAST-ADL2. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 6351 LNCS, 332–346 (2010).

200. Schmittner, C. et al.: Using SAE J3061 for automotive security requirement engineering. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 9923 LNCS, 157–170 (2016).

201. Schneider, D.: Conditional Safety Certification for Open Adaptive Systems (PhD Theses in Experimental Software Engineering). Fraunhofer Verlag (March 26, 2014), Kaiserslautern (2014).

202. Schneider, D., Becker, M.: Runtime Models for Self-Adaptation in the Ambient Assisted Living Domain. Proc. 3rd Intl. Work. Model. ii, 47–56 (2008).

203. Schneider, D., Trapp, M.: Conditional Safety Certification of Open Adaptive Systems. ACM Trans. Auton. Adapt. Syst. 8, 2, 1–20 (2013).

204. Schneider, D., Trapp, M.: Conditional Safety Certification of Open Adaptive Systems. ACM Trans. Auton. Adapt. Syst. 8, 2, 1–20 (2013).

205. Schoitsch, E.: Design for safety and security of complex embedded systems: A unified approach. NATO Adv. Res. Work. Cybersp. Secur. Def. Res. Issues. 161–174 (2004).

206. Selvaraj, S., Sundaravaradhan, S.: Challenges and opportunities in IoT healthcare systems: a systematic review. SN Appl. Sci. 2, 1, 1–8 (2020).

207. Sendelbach, S., Funk, M.: Alarm Fatigue. AACN Adv. Crit. Care. 24, 4, 378–386 (2013).

208. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976).

209. Shahrdar, S. et al.: A Survey on Trust in Autonomous Systems. In: Arai, K. et al. (eds.) Intelligent Computing. pp. 368–386 Springer International Publishing, Cham (2019).

210. Shepherd, C.H. et al.: Risk monitors - The State of the Art in their Development and Use at Nuclear Power Plants - Produced on behalf of IAEA and OECD/NEA WGRisk, https://inis.iaea.org/search/search.aspx?orig_q=RN:41041114, (2004).

211. Shi, J. et al.: A survey of Cyber-Physical Systems. In: 2011 International Conference on Wireless Communications and Signal Processing (WCSP). pp. 1–6 IEEE (2011).

212. Silva, L.C. et al.: A model-based architecture for testing medical Cyber-Physical Systems. 29th Annu. ACM Symp. Appl. Comput. SAC 2014. 25–30 (2014).

213. Simpao, A.F. et al.: The reliability of manual reporting of clinical events in an anesthesia information management system (AIMS). J. Clin. Monit. Comput. 26, 6, 437–439 (2012).

214. Sklet, S.: Safety barriers: Definition, classification, and performance. J. Loss Prev. Process Ind. 19, 5, 494–506 (2006).

215. Smith, C., Kelly, D.: Bayesian Inference for Probabilistic Risk Assessment. (2006).

216. Sokolsky, L.I.O.: Medical Cyber Physical Systems. Control. 743–748 (2010).

217. Srotyr, M. et al.: Pilot applications of cooperative systems. In: 2016 Smart Cities Symposium Prague, SCSP 2016. Institute of Electrical and Electronics Engineers Inc. (2016).

218. ST, L.: Crying wolf: false alarms in a pediatric intensive care unit. Crit Care Med. 22, 6, 981–985.

219. Stankovic, J.A.: Research directions for cyber physical systems in wireless and

mobile healthcare. ACM Trans. Cyber-Physical Syst. 1, 1, 1–12 (2017).

220. Stevens, N. et al.: Smart alarms: Multivariate medical alarm integration for post CABG surgery patients. In: IHI'12 - Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium. pp. 533–541 (2012).

221. Stevens, N. et al.: Smart Alarms: Multivariate Medical Alarm Integration for Post CABG Surgery Patients. In: Proceedings of the 2nd ACM SIGHIT symposium on International health informatics - IHI '12. p. 533 ACM Press, New York, New York, USA (2012).

222. Sujan, M. et al.: Critical Barriers to Safety Assurance and Regulation of Autonomous Medical Systems. In: Proceedings of the 29th European Safety and Reliability Conference (ESREL). pp. 4257–4262 Research Publishing Services, Singapore (2019).

223. Sujan, M.A. et al.: Should healthcare providers do safety cases? Lessons from a cross-industry review of safety case practices. Saf. Sci. 84, 181–189 (2016).

224. Summers, A.E.: Introduction to layers of protection analysis. J. Hazard. Mater. 104, 1–3, 163–168 (2003).

225. Sutton, R.T. et al.: An overview of clinical decision support systems: benefits, risks, and strategies for success, https://doi.org/10.1038/s41746-020-0221-y, (2020).

226. Tang, L.A. et al.: Tru-alarm: Trustworthiness analysis of sensor networks in cyber-physical systems. Proc. - IEEE Int. Conf. Data Mining, ICDM. 1079–1084 (2010).

227. Thieme, C.A. et al.: A Risk Management Framework for Unmanned Underwater Vehicles Focusing on Human and Organizational Factors. In: Volume 3: Structures, Safety and Reliability. p. V003T02A075 ASME (2015).

228. Thieme, C.A., Utne, I.B.: A risk model for autonomous marine systems and operation focusing on human–autonomy collaboration. Proc. Inst. Mech. Eng. Part O J. Risk Reliab. 231, 4, 446–464 (2017).

229. Thuemmler, C., Bai, C. eds: Health 4.0: How Virtualization and Big Data are Revolutionizing Healthcare. Springer International Publishing, Cham (2017).

230. Toben, T. et al.: Safe autonomous transport vehicles in heterogeneous outdoor environments. Commun. Comput. Inf. Sci. 336 CCIS, 61–75 (2012).

231. Trapp, M. et al.: A safety roadmap to cyber-physical systems. In: Münch, J. and

Schmid, K. (eds.) Perspectives on the Future of Software Engineering: Essays in Honor of Dieter Rombach. pp. 81–94 Springer Berlin Heidelberg, Berlin, Heidelberg (2013).

232. Trapp, M. et al.: A safety roadmap to cyber-physical systems. Perspect. Futur. Softw. Eng. Essays Honor Dieter Rombach. 81–94 (2013).

233. Trapp, M., Schneider, D.: Safety assurance of open adaptive systems - A survey. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 8378 LNCS, 279–318 (2014).

234. Vagia, M. et al.: A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed? Appl. Ergon. 53, 190–202 (2016).

235. Venkatasubramanian, K.K. et al.: Functional Alarms for Systems of Interoperable Medical Devices. In: Proceedings - 2014 IEEE 15th International Symposium on High-Assurance Systems Engineering, HASE 2014. pp. 247–248 IEEE (2014).

236. Venkatasubramanian, K.K. et al.: Requirement engineering for functional alarm system for interoperable medical devices. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 252–266 Springer, Cham (2015).

237. Wang, A. et al.: Selective and compressive sensing for energy-efficient implantable neural decoding. In: IEEE Biomedical Circuits and Systems Conference: Engineering for Healthy Minds and Able Bodies, BioCAS 2015 - Proceedings. Institute of Electrical and Electronics Engineers Inc. (2015).

238. Wardziński, A.: Safety assurance strategies for autonomous vehicles. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 277–290 Springer Berlin Heidelberg, Berlin, Heidelberg (2008).

239. Wei, K. et al.: Health Monitoring Based on Internet of Medical Things: Architecture, Enabling Technologies, and Applications. IEEE Access. 8, 27468–27478 (2020).

240. Wei, R. et al.: DEIS: Dependability Engineering Innovation for Cyber-Physical Systems. Presented at the (2018).

241. Weimer, J. et al.: Parameter-invariant monitor design for cyber-physical systems. Proc. IEEE. 106, 1, 71–92 (2018).

242. Weiss, G. et al.: Safe adaptation for reliable and energy-efficient E/E architectures. In: SpringerBriefs in Applied Sciences and Technology. pp. 1–18 (2018).

243. Wilkinson, P.J., Kelly, T.P.: Functional hazard analysis for highly integrated aerospace systems. Certif. Ground/Air Syst. Semin. (Ref. No. 1998/255), IEE. 1–4 (1998).

244. Winner, H.: Challenges of automotive systems engineering for industry and academia. Automot. Syst. Eng. 3–15 (2013).

245. Winner, H., Schopper, M.: Adaptive Cruise Control. In: Handbook of Driver Assistance Systems. pp. 1093–1148 Springer International Publishing, Cham (2016).

246. Yan, Y. et al.: A restricted Boltzmann machine based two-lead electrocardiography classification. In: 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2015. Institute of Electrical and Electronics Engineers Inc. (2015).

247. Yelderman, M., New, W.: Evaluation of pulse oximetry. Anesthesiology. 59, 4, 349–352 (1983).

248. Zeng, Z., Zio, E.: Dynamic Risk Assessment Based on Statistical Failure Data and Condition-Monitoring Degradation Data. IEEE Trans. Reliab. 67, 2, 609–622 (2018).

249. Zhang, Y., Ji, Q.: Active and dynamic information fusion for multisensor systems with dynamic Bayesian networks. IEEE Trans. Syst. Man. Cybern. B. Cybern. 36, 2, 467–472 (2006).

250. Zio, E.: Critical Infrastructures Vulnerability and Risk Analysis. Eur. J. Secur. Res. 1, 2, 97–114 (2016).

251. Zio, E.: The future of risk assessment. Reliab. Eng. Syst. Saf. 177, March, 176–190 (2018).

252. BS EN 61508-1 : 2010 BSI Standards Publication Functional safety of electrical / electronic / programmable electronic safety-related systems Part 1 : General requirements. (2010).

253. BS EN 61508-4 : 2010 BSI Standards Publication Functional safety of electrical / electronic / programmable electronic safety related systems Part 4 : Definitions and abbreviations. (2010).

254. BS EN 61508-5 : 2010 BSI Standards Publication Functional safety of electrical / electronic / programmable electronic safety related systems Part 5 : Examples of methods for the determination. (2010).

255. IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE, or E/E/PES). (2010).

Curriculum Vitae

Name	Fábio Luiz Leite Júnior	

Education:		
Ph.D. in Computer Science	2015-2021	Technische Universität Kaiserslautern
Master's Degree in Computer Science	2005-2007	Universidade Federal de Campina Grande, Campina Grande, Brasil
Bachelor's Degree in Computer Science	2000-2005	Universidade Federal de Campina Grande, Campina Grande, Brasil

Current Position:	2010-present	Assistant Professor at the Department of Computer Science at Universidade Estadual da Paraíba, Brasil

Kaiserslautern, 19. October 2021